Truth-Functional Logic

Syntax

Every atomic sentence (A, B, C, …) is a sentence

\( \top \) and \( \bot \) are sentences

With \( \varphi \) a sentence, the negation \( \neg \varphi \) is a sentence

With \( \varphi \) and \( \psi \) sentences, the conjunction \( \varphi \land \psi \) is a sentence (\( \varphi \) and \( \psi \) are called conjuncts)

With \( \varphi \) and \( \psi \) sentences, the disjunction \( \varphi \lor \psi \) is a sentence (\( \varphi \) and \( \psi \) are called disjuncts)

With \( \varphi \) and \( \psi \) sentences, the conditional \( \varphi \to \psi \) is a sentence (\( \varphi \) is the antecedent, \( \psi \) the consequent)

With \( \varphi \) and \( \psi \) sentences, the biconditional \( \varphi \leftrightarrow \psi \) is a sentence

All sentences that are not atomic are compound sentences.

A literal is either an atomic sentence or the negation of an atomic sentence.

The complement \( \mathcal{L}' \) of a literal \( \mathcal{L} \) is:

1) \( \neg P \) if \( \mathcal{L} = P \)
2) \( P \) if \( \mathcal{L} = \neg P \)

where \( P \) is an atomic sentence

In the conjunction \( \varphi \land \psi \), the \( \land \) by itself is called a logical connective, as it connects (it ‘conjunction’, in this case) two component sentences. Likewise, \( \lor \), \( \to \), and \( \leftrightarrow \) are logical connectives. In fact, even \( \neg \) is a connective, even though it only ‘connects’ one sentence. The connectives \( \land, \lor, \neg, \to, \leftrightarrow \) are referred to as conjunction, disjunction, negation, conditional, and biconditional, respectively. The connectives \( \land, \lor, \neg \) and \( \to \) are Boolean connectives, named after the 19th century founder of modern formal logic, George Boole. The part of truth-functional logic that deals with the Boolean connectives only is called Boolean Logic.
Parentheses and the Method of Induction

A statement like $A \land B \lor C$ is ambiguous: is this the conjunction of $A$ and $B \lor C$, or is this a disjunction of $A \land B$ and $C$? We will require that our statements are not ambiguous, and we will use parentheses to do so. So, if this statement was meant to be a conjunction of $A$ and $B \lor C$, we will write $A \land (B \lor C)$.

In fact, some textbooks will make parentheses simply part of the formal syntax. That is, in the recursive definition of syntactical expressions that we saw on the previous page, they will define a conjunction of $A$ and $B$ as $(A \land B)$, and the same for disjunctions, conditionals, and biconditionals. So, the sentence $A \land (B \lor C)$ will actually end up as $(A \land (B \lor C))$! We will keep things a little more informal, especially since this business with parentheses is all completely intuitive and obvious.

In fact, let’s prove the following, completely obvious, theorem about sentences of our language:

Theorem: Every sentence has just as many left parentheses as right parentheses

The proof of this theorem is only interesting in so far that it demonstrates an important technique to prove claims about sentences. This technique is mathematical induction. In particular, to prove that every sentence has some property $P$, we will prove two things:

A. Inductive Base: Every atomic sentence has property $P$
B. Inductive Step: For any sentences $\varphi$ and $\psi$:
   
   - If $\varphi$ and $\psi$ have property $P$ (this is the Inductive Hypothesis or Inductive Assumption), then $\neg \varphi$, $\varphi \land \psi$, $\varphi \lor \psi$, $\varphi \rightarrow \psi$, and $\varphi \leftrightarrow \psi$ have property $P$ as well.

It should be obvious, given the recursive definition of sentences, that from these two things, it follows that indeed all sentences have property $P$.

So, back to the Theorem. Here is the Proof:

A. Inductive Base: All atomic sentences do indeed have just as many left and right parentheses, namely 0.
B. Inductive Step: assuming $\varphi$ and $\psi$ both have just as many left and right parentheses, $\neg \varphi$, $\varphi \land \psi$, $\varphi \lor \psi$, $\varphi \rightarrow \psi$, and $\varphi \leftrightarrow \psi$ will all have just as many left and right parentheses as well (here we could say something like: if $\varphi$ has $m$ left and right parentheses, and $\psi$ has $n$ left and right parentheses, then $\varphi \land \psi$ has $m + n$ left and right parentheses (plus maybe a few more for disambiguation), but you get the point)

That’s it!
Semantics

Where \( L \) is the set of all syntactical sentences, a *truth-assignment* \( h \) is a function \( h:L \rightarrow \{\text{True}, \text{False}\} \) that satisfies the following conditions:

\[
\begin{align*}
h(\top) &= \text{True} \\
h(\bot) &= \text{False} \\
\end{align*}
\]

\[
\begin{align*}
h(\neg \varphi) &= \text{True } \text{iff } h(\varphi) = \text{False} \\
h(\varphi \land \psi) &= \text{True } \text{iff } h(\varphi) = \text{True} \text{ and } h(\psi) = \text{True} \\
h(\varphi \lor \psi) &= \text{False } \text{iff } h(\varphi) = \text{False} \text{ and } h(\psi) = \text{False} \\
h(\varphi \rightarrow \psi) &= \text{False } \text{iff } h(\varphi) = \text{True} \text{ and } h(\psi) = \text{False} \\
h(\varphi \leftrightarrow \psi) &= \text{True } \text{iff } h(\varphi) = \text{True} \text{ and } h(\psi) = \text{True} \text{ or } h(\varphi) = \text{False} \text{ and } h(\psi) = \text{False}
\end{align*}
\]

We thus see that truth-assignments assign truth-values to sentences in such a way that the truth-value of a compound sentence is systematically related to the truth-values of its component parts. More specifically, given any truth-assignment \( h \), the truth-value that \( h \) assigns to a compound sentence can be seen as a *function* of the truth-values that \( h \) assigns to its component parts. Indeed, the logical connectives (negation, conjunction, disjunction, conditional and biconditional) can be seen as *truth-functions*: functions that take in some number of truth-values (1 in the case of negation, and 2 in the case of conjunction, disjunction, conditional, and biconditional) and output a new truth-value. That’s why the logical connectives are also often called *logical operators*. And, it is why sentential logic, or propositional logic, is called “Truth-Functional” Logic.
Truth-Tables

Truth-Tables can be used to depict truth-functional semantics:

Negation:

<table>
<thead>
<tr>
<th>( \varphi )</th>
<th>( \neg \varphi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Conjunction, Disjunction, Conditional, Biconditional:

<table>
<thead>
<tr>
<th>( \varphi )</th>
<th>( \psi )</th>
<th>( \varphi \lor \psi )</th>
<th>( \varphi \land \psi )</th>
<th>( \varphi \rightarrow \psi )</th>
<th>( \varphi \leftrightarrow \psi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Truth-tables can also be used to find the truth-conditions of any compound sentence.

Example:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>( A \land (B \lor C) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Tautology, Contradiction, Contingency

A sentence ϕ is a tautology iff there is no truth-function h such that h(ϕ) = False

Example:
A ∨ ¬A is a tautology:

<table>
<thead>
<tr>
<th>A</th>
<th>A ∨ ¬A</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

A sentence ϕ is a contradiction iff there is no truth-function h such that h(ϕ) = True

Example:
A ∧ ¬A is a tautology:

<table>
<thead>
<tr>
<th>A</th>
<th>A ∧ ¬A</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

A sentence ϕ is contingent iff there exists a truth-function h such that h(ϕ) = True and there exists a truth-function h such that h(ϕ) = False

Example:
A is contingent:

<table>
<thead>
<tr>
<th>A</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Equivalence, Contradictory, Contrary, Subcontrary, Implication

Two sentences $\varphi$ and $\psi$ are equivalent iff for all $h$: $h(\varphi) = \text{True}$ iff $h(\psi) = \text{True}$

Example:

A and $A \lor A$ are equivalent:

<table>
<thead>
<tr>
<th>A</th>
<th>$A \lor A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

When $\varphi$ and $\psi$ are equivalent we write: $\varphi \leftrightarrow \psi$

Two sentences $\varphi$ and $\psi$ are contradictory iff for all $h$: $h(\varphi) = \text{True}$ iff $h(\psi) = \text{False}$

Two sentences $\varphi$ and $\psi$ are contrary iff there exists no $h$ such that $h(\varphi) = \text{True}$ and $h(\psi) = \text{True}$

Two sentences $\varphi$ and $\psi$ are subcontrary iff there exists no $h$ such that $h(\varphi) = \text{False}$ and $h(\psi) = \text{False}$

A sentence $\varphi$ implies sentence $\psi$ iff there exists no $h$ such that $h(\varphi) = \text{True}$ and $h(\psi) = \text{False}$

Example:

A implies $A \lor B$:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>$A \lor B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

When $\varphi$ implies $\psi$ we write: $\varphi \Rightarrow \psi$

When $\varphi$ implies $\psi$ we say that $\varphi$ is a stronger statement than $\psi$ or, conversely, that $\psi$ is a weaker statement than $\varphi$. Sometimes, we say that $\psi$ is subsumed by $\varphi$. This is most easily understood with an example where $\varphi = P \land Q$ and $\psi = P$; since $P$ is part of what is already asserted in $P \land Q$, we say that $P$ has been subsumed by $P \land Q$. 
Implication, Consistency, Validity

A set of sentences $\Gamma = \{\varphi_1, \ldots, \varphi_n\}$ implies a sentence $\psi$ iff there exists no $h$ such that $h(\varphi_i) = \text{True}$ for all sentences $\varphi_i$ in $\Gamma$ and $h(\psi) = \text{False}$

When $\Gamma$ implies $\psi$ we write $\Gamma \models \psi$

We can write $\varphi_1, \ldots, \varphi_n \models \psi$ as short for $\{\varphi_1, \ldots, \varphi_n\} \models \psi$.

Similarly, $\Gamma, \varphi \models \psi$ is short for $\Gamma \cup \{\varphi\} \models \psi$, and $\Gamma, \Delta \models \psi$ is short for $\Gamma \cup \Delta \models \psi$

Theorem 1.0.1: $\varphi \Rightarrow \psi$ iff $\varphi \models \psi$

A set of sentences $\Gamma = \{\varphi_1, \ldots, \varphi_n\}$ is consistent iff there exists an $h$ such that $h(\varphi_i) = \text{True}$ for all sentences $\varphi_i$ in $\Gamma$

When a set of sentences $\Gamma$ is inconsistent, we write $\Gamma \models$

Theorem 1.0.2: $\Gamma \models$ iff $\Gamma \models \bot$

Proof:

$\Gamma \models$ iff (def. $\models$)

there exists no $h$ such that $h(\varphi) = \text{True}$ for all sentences in $\Gamma$ iff (sem. $\bot$)

there exists no $h$ such that $h(\varphi) = \text{True}$ for all sentences in $\Gamma$ and $h(\bot) = \text{False}$ iff (def. $\models$)

$\Gamma \models \bot$

When $\Gamma$ implies $\psi$ and $\Gamma = \emptyset$ (the empty set), we write $\models \psi$

A sentence $\psi$ is valid iff $\models \psi$.

Theorem 1.0.3: $\psi$ is a tautology iff $\psi$ is valid

An argument $\varphi_1, \ldots, \varphi_n / \therefore \psi$ is valid iff there exists no $h$ such that $h(\varphi_i) = \text{True}$ for all sentences $\varphi_i$ and $h(\psi) = \text{False}$

Theorem 1.0.4: An argument $\varphi_1, \ldots, \varphi_n / \therefore \psi$ is valid iff $\{\varphi_1, \ldots, \varphi_n\} \models \psi$

Proofs of any unproven theorems above are trivial and left to the reader.
Some Theorems regarding Tautology, Contradiction, and Contingency

Theorem 1.1.1: \( \varphi \) is a tautology iff \( \neg \varphi \) is a contradiction

Proof:

\( \varphi \) is a tautology iff (definition tautology)
there is no \( h \) such that \( h(\varphi) = \text{False} \) iff (semantics \( \neg \))
there is no \( h \) such that \( h(\neg \varphi) = \text{True} \) iff (definition contradiction)
\( \neg \varphi \) is a contradiction

Theorem 1.1.2: \( \varphi \) is a contradiction iff \( \neg \varphi \) is a tautology

Proof:

\( \varphi \) is a contradiction iff (definition contradiction)
there is no \( h \) such that \( h(\varphi) = \text{True} \) iff (semantics \( \neg \))
there is no \( h \) such that \( h(\neg \varphi) = \text{False} \) iff (definition tautology)
\( \neg \varphi \) is a tautology

Theorem 1.1.3: \( \varphi \) is contingent iff \( \neg \varphi \) is contingent

Proof:

\( \varphi \) is contingent iff (def. contingency)
there exists an \( h \) such that \( h(\varphi) = \text{True} \) and there exists an \( h \) such that \( h(\varphi) = \text{False} \) iff (sem. \( \neg \))
there exists an \( h \) such that \( h(\neg \varphi) = \text{False} \) and there exists an \( h \) such that \( h(\neg \varphi) = \text{True} \) iff (def. contingency)
\( \neg \varphi \) is contingent

Theorem 1.1.4: For any statement \( \varphi \): \( \bot \Rightarrow \varphi \) (i.e. a contradiction implies anything)

Theorem 1.1.5: For any statement \( \varphi \): \( \varphi \Rightarrow \top \) (i.e. anything implies a tautology)

The last two theorems reveal a useful way to think about \( \bot \) (contradiction) and \( \top \) (tautology). Basically, a contradiction claims everything (i.e. that pigs fly, that pigs don’t fly, that \( 1+1 = 2 \), that \( 1+1 = 3 \), etc), while a tautology claims nothing at all. This means that there is no information contained in a tautology (indeed, to say that “it will rain or it won’t” doesn’t tell us anything at all). Of course, by claiming that everything is the case, a contradiction isn’t very helpful either!
Some Theorems regarding Equivalence, Implication, Contradictory, Contrary, and Subcontrary

Theorem 1.2.1: $\phi \iff \psi$ iff $\phi \Rightarrow \psi$ and $\psi \Rightarrow \phi$

Proof:

$\phi \iff \psi$ iff (def. $\iff$)
for all $h$: $h(\phi) = \text{True}$ iff $h(\psi) = \text{True}$ iff (exhaust only two possibilities)
there is no $h$ such that $h(\phi) = \text{True}$ and $h(\psi) = \text{False}$ and there is no $h$ such that $h(\psi) = \text{True}$ and $h(\phi) = \text{False}$ iff (def. $\Rightarrow$)
$\phi \Rightarrow \psi$ and $\psi \Rightarrow \phi$

Theorem 1.2.2: Two statements $\phi$ and $\psi$ are contradictory iff $\phi$ and $\psi$ are contrary and subcontrary.

Proof:

$\phi$ and $\psi$ are contradictory iff
it is impossible for $\phi$ and $\psi$ to have the same truth-value iff
it is impossible for $\phi$ and $\psi$ to be true and it is impossible for $\phi$ and $\psi$ to be false iff
$\phi$ and $\psi$ are contrary and subcontrary.

Theorem 1.2.3: $\phi \iff \psi$ iff $\phi$ and $\neg \psi$ are contradictory

Proof: Left to the reader.

Theorem 1.2.4: If $\phi \Rightarrow \psi$ and $\psi \Rightarrow \lambda$, then $\phi \Rightarrow \lambda$

Proof:

Assume $\phi \Rightarrow \psi$ and $\psi \Rightarrow \lambda$. Then 1) there exists no $h$ such that $h(\phi) = \text{True}$ and $h(\psi) = \text{False}$, and 2) there exists no $h$ such that $h(\psi) = \text{True}$ and $h(\lambda) = \text{False}$. So, if we take any $h$ for which $h(\phi) = \text{True}$, then by 1) it must also be the case that $h(\psi) = \text{True}$, and hence by 2) it must also be the case that $h(\lambda) = \text{True}$. Therefore, there cannot exist an $h$ such that $h(\phi) = \text{True}$ and $h(\lambda) = \text{False}$. Therefore, $\phi \Rightarrow \lambda$.

Theorem 1.2.5: $\phi \Rightarrow \psi$ iff $\neg \psi \Rightarrow \neg \phi$

Theorem 1.2.6: $\top \iff \neg \bot$

Theorem 1.2.7: $\phi \land \neg \phi \Rightarrow \bot$

Proofs: Left to the reader.
Some Theorems regarding Consistency, Implication, and Validity

Theorem 1.3.1: If $\varphi \in \Gamma$ then $\Gamma \models \varphi$

Proof: If $\varphi \in \Gamma$, then for any $h$ such that $h(\psi) = \text{True}$ for all sentences $\psi$ in $\Gamma$, $h(\varphi) = \text{True}$. Hence, by definition, $\Gamma \models \varphi$.

Special Case (Assumptions Principle): $\{\varphi\} \models \varphi$ (or just $\varphi \models \varphi$)

Theorem 1.3.2: If $\Gamma_1 \models \psi$ and $\Gamma_1 \subseteq \Gamma_2$ then $\Gamma_2 \models \psi$

Special Case (Thinning Principle): If $\Gamma \models \varphi$ then $\Gamma, \psi \models \varphi$

Theorem 1.3.3 (Cutting Principle): If $\Gamma \models \varphi$ and $\Delta, \varphi \models \psi$ then $\Gamma, \Delta \models \psi$

Special Case: If $\Gamma \models \varphi$ and $\varphi \Rightarrow \psi$ then $\Gamma \models \psi$

Theorem 1.3.4 (Negation Basic Principle): $\Gamma \models \psi$ iff $\Gamma \cup \{\neg \psi\} \models$

Proof:

$\Gamma \models \psi$ iff (def. $\models$) there exists no $h$ s.t. $h(\varphi) = \text{True}$ for all sentences $\varphi$ in $\Gamma$ and $h(\psi) = \text{False}$ iff (sem. $\neg$) there exists no $h$ s.t. $h(\varphi) = \text{True}$ for all sentences $\varphi$ in $\Gamma$ and $h(\neg \psi) = \text{True}$ iff (same) there exists no $h$ s.t. $h(\varphi) = \text{True}$ for all sentences $\varphi$ in $\Gamma \cup \{\neg \psi\}$ iff (def. $\models$) $\Gamma \cup \{\neg \psi\} \models$

Theorem 1.3.5 (Conjunction Basic Principle): $\Gamma \models \varphi \land \psi$ iff $\Gamma \models \varphi$ and $\Gamma \models \psi$

Proof: Left to reader

Theorem 1.3.6 (Disjunction Basic Principle): $\Gamma, \varphi \lor \psi \models \text{iff} \Gamma, \varphi \models$ and $\Gamma, \psi \models$

Proof: Homework 1

Theorem 1.3.7 (Conditional Basic Principle): $\Gamma \models \varphi \Rightarrow \psi$ iff $\Gamma, \varphi \models \psi$

Proof: Left to Reader
Some Theorems regarding Tautology, Contradiction, and Implication

Theorem 1.4.1: ψ is a tautology iff ⊨ ψ

Proof:

ψ is a tautology iff (def. tautology)
there is no h such that h(ψ) = False iff (no sentences in ∅)
there is no h such that h(ϕ) = True for all sentences ϕ in ∅ and h(ψ) = False iff (def. ⊨)
⊨ ψ

Theorem 1.4.2: ψ is a tautology iff ⊤ ⇒ ψ

Proof:

ψ is a tautology iff (def. tautology)
there is no h such that h(ψ) = False iff (sem. ⊤)
there is no h such that h(⊤) = True and h(ψ) = False iff
⊤ ⇒ ψ

Theorem 1.4.3: φ is a contradiction iff ⊨ ¬φ

Proof:

φ is a contradiction iff (Theorem 1.1.2)
¬φ is a tautology iff (Theorem 1.4.1)
⊨ ¬φ

Theorem 1.4.4: φ is a contradiction iff φ ⇒ ⊥

Proof:

φ is a contradiction iff (1.1.2)
¬φ is a tautology iff (1.4.2)
⊤ ⇒ ¬φ iff (1.2.6, 1.2.1, and 1.2.4)
¬⊥ ⇒ ¬φ iff (1.2.5)
φ ⇒ ⊥

Theorem 1.4.5: φ → ψ is a tautology iff φ ⇒ ψ

Proof: Left to Reader
Some Important Equivalences

Boolean Logic Equivalences:

Double Negation:
1) \( \varphi \Leftrightarrow \neg
\neg \varphi \)

Association:
1) \( \varphi \land (\psi \land \lambda) \Leftrightarrow (\varphi \land \psi) \land \lambda \)
2) \( \varphi \lor (\psi \lor \lambda) \Leftrightarrow (\varphi \lor \psi) \lor \lambda \)

Commutation:
1) \( \varphi \land \psi \Leftrightarrow \psi \land \varphi \)
2) \( \varphi \lor \psi \Leftrightarrow \psi \lor \varphi \)

Idempotence:
1) \( \varphi \land \varphi \Leftrightarrow \varphi \)
2) \( \varphi \lor \varphi \Leftrightarrow \varphi \)

DeMorgan:
1) \( \neg (\varphi \land \psi) \Leftrightarrow \neg \varphi \lor \neg \psi \)
2) \( \neg (\varphi \lor \psi) \Leftrightarrow \neg \varphi \land \neg \psi \)

Distribution:
1) \( \varphi \land (\psi \lor \lambda) \Leftrightarrow (\varphi \land \psi) \lor (\varphi \land \lambda) \)
2) \( \varphi \lor (\psi \land \lambda) \Leftrightarrow (\varphi \lor \psi) \land (\varphi \lor \lambda) \)

Absorption:
1) \( \varphi \land (\varphi \lor \psi) \Leftrightarrow \varphi \)
2) \( \varphi \lor (\varphi \land \psi) \Leftrightarrow \varphi \)

The Absorption principle is sometimes referred to as the Subsumption principle (at least the first half is often called Subsumption). Indeed, the Absorption principle can be seen as a specific case of what happens when we conjunct or disjunct two statements, where one subsumes the other: the conjunction is equivalent to the subsuming statement, while the disjunction is equivalent to the subsumed statement.

Reduction:
1) \( \varphi \land (\neg \varphi \lor \psi) \Leftrightarrow \varphi \land \psi \)
2) \( \varphi \lor (\neg \varphi \land \psi) \Leftrightarrow \varphi \lor \psi \)

The first half of the Reduction principle is most easily grasped: Given that \( \varphi \) definitely has to be true, we can ‘reduce’ the second term, \( \neg \varphi \lor \psi \), to just \( \psi \). The Subsumption and Reduction principles are very useful for simplifying statements that are in CNF or DNF (see later).
Some important laws involving the conditional and biconditional:

Contraposition:
1) \( \varphi \rightarrow \psi \iff \neg \psi \rightarrow \neg \varphi \)

Exportation:
1) \( \varphi \rightarrow (\psi \rightarrow \lambda) \iff (\varphi \land \psi) \rightarrow \lambda \)

Commutation of \( \iff \):
1) \( \varphi \iff \psi \iff \psi \iff \varphi \)

Association of \( \iff \):
1) \( \varphi \iff (\psi \iff \lambda) \iff (\varphi \iff \psi) \iff \lambda \)

Some Laws to rewrite Non-Boolean Connectives in terms of Boolean Connectives:

Implication:
1) \( \varphi \rightarrow \psi \iff \neg \varphi \lor \psi \)
2) \( \varphi \rightarrow \psi \iff \neg (\varphi \land \neg \psi) \)

Equivalence:
1) \( \varphi \iff \psi \iff (\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi) \) (and thus \( \iff (\neg \varphi \lor \psi) \land (\neg \psi \lor \varphi) \))
2) \( \varphi \iff \psi \iff (\varphi \land \psi) \lor (\neg \varphi \land \neg \psi) \)

Some Laws regarding \( \top \) and \( \bot \):
1) \( P \land \neg P \iff \bot \)
2) \( P \lor \neg P \iff \top \)
3) \( \neg \bot \iff \top \)
4) \( \neg \top \iff \bot \)
5) \( P \land \bot \iff \bot \)
6) \( P \lor \top \iff \top \)
7) \( P \land \top \iff P \)
8) \( P \lor \bot \iff P \)
9) \( P \rightarrow \top \iff \top \)
10) \( P \rightarrow \bot \iff \neg P \)
11) \( \top \rightarrow P \iff P \)
12) \( \bot \rightarrow P \iff \top \)
13) \( P \iff \top \iff P \)
14) \( P \iff \bot \iff \neg P \)
Generalized Conjunctions and Disjunctions

Because of the Association equivalences for conjunctions and disjunctions, we can write an expression like $\varphi \land (\psi \land \lambda)$ simply as $\varphi \land \psi \land \lambda$, and treat the latter as a conjunction with 3 conjuncts. Of course, the same can be done for more than 3 conjuncts as well, and the same goes for disjunctions. Thus, we reach the notion of a generalized conjunction (a conjunction of 0 or more conjuncts) and of a generalized disjunction (a disjunction of 0 or more disjuncts).

The semantics for generalized conjunctions and disjunctions is as follows:

$h(\varphi_1 \land \ldots \land \varphi_n) = \text{True iff } h(\varphi_i) = \text{True for all } \varphi_i.$

$h(\varphi_1 \lor \ldots \lor \varphi_n) = \text{True iff } h(\varphi_i) = \text{True for some } \varphi_i.$

Thus we see that where $\varphi$ is a generalized conjunction with 0 conjuncts, $h(\varphi) = \text{True for any } h$, and hence we will simply treat this as $\top$. Indeed, this is consistent with the earlier finding that a tautology says nothing at all. Similarly, where $\varphi$ is a generalized disjunction with 0 disjuncts, $h(\varphi) = \text{False for any } h$, and hence we will simply treat this as $\bot$.

Many of the equivalence principles can be generalized to account for generalized conjunctions and disjunctions. For example:

Generalized Commutation:
1) $\varphi_1 \land \ldots \land \varphi_n \iff (\text{all the } \varphi_i \text{'s conjuncted in any order})$
2) $\varphi_1 \lor \ldots \lor \varphi_n \iff (\text{all the } \varphi_i \text{'s disjuncted in any order})$

Generalized Idempotence:
1) $\varphi \land \ldots \land \varphi \iff \varphi$
2) $\varphi \lor \ldots \lor \varphi \iff \varphi$

Generalized DeMorgan:
1) $\neg(\varphi_1 \land \ldots \land \varphi_n) \iff \neg \varphi_1 \lor \ldots \lor \neg \varphi_n$
2) $\neg(\varphi_1 \lor \ldots \lor \varphi_n) \iff \neg \varphi_1 \land \ldots \land \neg \varphi_n$

Generalized Distribution:
1) $\varphi \land (\varphi_1 \land \ldots \land \varphi_n) \iff (\varphi \land \varphi_1) \land \ldots \land (\varphi \land \varphi_n)$
2) $\varphi \lor (\varphi_1 \land \ldots \land \varphi_n) \iff (\varphi \lor \varphi_1) \lor \ldots \lor (\varphi \lor \varphi_n)$
Principle of Substitution of Logical Equivalents

Let us write $S(\phi)$ for a sentence that has $\phi$ as one of its component parts and $S(\psi)$ for the sentence that results when substituting $\psi$ for $\phi$.

Theorem: If $\phi \iff \psi$, then $S(\phi) \iff S(\psi)$ (Principle of Substitution of Equivalents)

Proof: (by induction on the formation of $S$)

First, five Lemma’s:

Lemma Subst. 1: If $\phi \iff \psi$, then $\neg \phi \iff \neg \psi$
Proof: Follows from 1.2.1 and 1.2.5.

Lemma Subst. 2: If $\phi \iff \psi$, then for any $\lambda$: $\phi \land \lambda \iff \psi \land \lambda$ and $\lambda \land \phi \iff \lambda \land \psi$
Proof: If $\phi \iff \psi$, then for any $h$: $h(\phi)$ = True iff $h(\psi)$ = True. So, if $\phi \iff \psi$, then for any $\lambda$: $h(\phi \land \lambda) =$ True iff $h(\phi) =$ True and $h(\lambda) =$ True iff $h(\psi) =$ True and $h(\lambda) =$ True iff $h(\psi \land \lambda) =$ True. Hence, if $\phi \iff \psi$, then $\phi \land \lambda \iff \psi \land \lambda$. The other half is completely analogous.

Similarly, the following Lemma’s can be proven:

Lemma Subst. 3: If $\phi \iff \psi$, then for any $\lambda$: $\phi \lor \lambda \iff \psi \lor \lambda$ and $\lambda \lor \phi \iff \lambda \lor \psi$
Lemma Subst. 4: If $\phi \iff \psi$, then for any $\lambda$: $\phi \rightarrow \lambda \iff \psi \rightarrow \lambda$ and $\lambda \rightarrow \phi \iff \lambda \rightarrow \psi$
Lemma Subst. 5: If $\phi \iff \psi$, then for any $\lambda$: $\phi \leftrightarrow \lambda \iff \psi \leftrightarrow \lambda$ and $\lambda \leftrightarrow \phi \iff \lambda \leftrightarrow \psi$

Now, assume $\phi \iff \psi$. We will then prove by induction that for any sentence $S$ that is the result of recursively applying any of the formation rules starting with $\phi$: $S(\phi) \iff S(\psi)$.

Induction Basis:

If $S = \phi$, then $S(\phi) \iff S(\psi)$.

Induction Step:

Suppose $S$ is the result of recursively applying any of the formation rules starting with $\phi$, and suppose that $S(\phi) \iff S(\psi)$. We now have to prove that if $S'$ is the application of one of the formation rules on $S$, then $S'(\phi) \iff S'(\psi)$. But this follows immediately from the five Lemma’s above. Hence, by mathematical induction, the principle follows.
Normal Forms

A sentence $\varphi$ is in *Negation Normal Form* (NNF) iff all negations that occur in $\varphi$ are negations of atomic sentences.

Any statement composed of conjunctions, disjunctions, and negations only can be ‘put into NNF’ through the repeated application of the Principle of Substitution of Logical Equivalents based on the DeMorgan and Double Negation equivalences.

Example:

$$\neg((A \lor B) \land \neg C) \leftrightarrow \text{(DeMorgan and Substitution Principle)}$$
$$\neg(A \lor B) \lor \neg C \leftrightarrow \text{(Double Negation and Substitution Principle)}$$
$$\neg(A \lor B) \lor C \leftrightarrow \text{(DeMorgan and Substitution Principle)}$$
$$(\neg A \land \neg B) \lor C \text{ (This is in NNF)}$$

A sentence $\varphi$ is in *Conjunctive Normal Form* (CNF) iff $\varphi$ is a generalized conjunction of generalized disjunctions of literals.

Any statement in NNF can be ‘put into CNF’ through the repeated application of the Principle of Substitution of Logical Equivalents based on the distribution of $\lor$ over $\land$.

Example:

$$(\neg A \land \neg B) \lor C \leftrightarrow$$
$$\neg A \lor C \land \neg B \lor C \text{ (This is in CNF)}$$

A sentence $\varphi$ is in *Disjunctive Normal Form* (DNF) iff $\varphi$ is a generalized disjunction of generalized conjunctions of literals.

Any statement in NNF can be ‘put into DNF’ through the repeated application of the Principle of Substitution of Logical Equivalents based on the distribution of $\land$ over $\lor$.

Example:

$$(A \lor B) \land (C \lor D) \leftrightarrow$$
$$[(A \lor B) \land C] \lor [(A \lor B) \land D] \leftrightarrow$$
$$(A \land C) \lor (B \land C) \lor (A \land D) \lor (B \land D) \text{ (This is in DNF)}$$
Truth-Functions

A truth-function is a function \( t: \{T,F\}^n \rightarrow \{T,F\} \) that takes in \( n \) truth-values and maps this into a new truth-value.

We have seen 4 binary truth-functions (truth-functions with 2 arguments): conjunction, disjunction, conditional, and biconditional. We have also encountered a unary connective: negation. And, we have encountered the only two truth-functions with 0 arguments: \( \top \) and \( \bot \).

All truth-functions can be depicted in a truth-table. In general, with \( n \) arguments, there are \( 2^n \) possible truth-value combinations to consider, so there will be \( k = 2^n \) rows in the truth-table. For every row, the function can map into true or false, and hence there are \( 2^k \) possible truth-function as defined over \( n \) arguments. For example, with 4 argument, there are \( 2^4=16 \) rows in the truth-table, giving us \( 2^{16}=65536 \) possible truth-functions for 4 arguments.

Below are all 4 unary truth-functions:

<table>
<thead>
<tr>
<th></th>
<th>( \top )</th>
<th>Id</th>
<th>( \neg )</th>
<th>( \bot )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

Here are all 16 binary ones:

<table>
<thead>
<tr>
<th></th>
<th>( \top )</th>
<th>( \lor )</th>
<th>( \leftarrow )</th>
<th>id(_1)</th>
<th>( \rightarrow )</th>
<th>id(_2)</th>
<th>( \leftrightarrow )</th>
<th>( \land )</th>
<th>nand</th>
<th>xor</th>
<th>nif</th>
<th>nid(_2)</th>
<th>nid(_1)</th>
<th>nonly</th>
<th>nor</th>
<th>( \bot )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

There are 256 ternary truth-functions. Below is just one of those. It can be used to capture ‘if … then … else’ sentences:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>If P, then Q, else R</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Expressive Completeness

While there are many truth-functions, we only use a few of them (namely, those that correspond to connectives in the English language). Fortunately, we can express many truth-functions by combining truth-functions that we already have. For example, \( P \text{ nand } Q \) is short for \( \neg(P \land Q) \). We can also easily verify that the ternary ‘if … then … else’ truth-function is captured by \( (P \land Q) \lor (\neg P \land R) \).

However, can we do this for all possible truth-functions, or are there some truth-functions whose truth-conditions cannot be expressed by the connectives that we have?

The answer is that we can express every truth-function in terms of \( \neg, \land, \lor \), and \( \leftrightarrow \). In fact, we can express every truth-function in terms of \( \neg, \land, \lor \) only, as can be easily demonstrated. Every truth-function can be depicted on a truth-table, and we can use a generalized conjunction to pick out every single row where the function evaluates to true. By making a generalized disjunction out of all these generalized conjunctions, we will have a statement that is equivalent to the original truth-function.

Example:

Using the above method, the ‘if … then … else’ truth-function can be captured by:
\[
(P \land Q \land R) \lor (P \land Q \land \neg R) \lor (\neg P \land Q \land R) \lor (\neg P \land \neg Q \land R)
\]

Because every truth-function can be expressed in terms of \( \neg, \land, \lor \), we say that the set \( \{\neg, \land, \lor\} \) is expressively complete.

Theorem: The sets \( \{\neg, \lor\} \) and \( \{\neg, \land\} \) are expressively complete.

Proof: Using DeMorgan, we can express \( \land \) in terms of \( \neg \) and \( \lor \), and we can express \( \lor \) in terms of \( \neg \) and \( \land \). So, since \( \{\neg, \land, \lor\} \) is expressively complete, \( \{\neg, \lor\} \) and \( \{\neg, \land\} \) are expressively complete as well.

Theorem: The sets \( \{\neg, \rightarrow\} \) and \( \{\rightarrow, \bot\} \) are expressively complete.

Proof: Left to the Reader

Theorem: The sets \( \{\text{nor}\} \) and \( \{\text{nand}\} \) are expressively complete.

Proof: This follows from the expressive completeness of \( \{\neg, \lor\} \) and \( \{\neg, \land\} \) and the following equivalences:
\[
\varphi \text{ nand } \varphi \leftrightarrow \neg((\varphi \land \varphi) \leftrightarrow \neg\varphi) \\
(\varphi \text{ nand } \varphi) \text{ nand } (\psi \text{ nand } \psi) \leftrightarrow \neg((\varphi \text{ nand } \varphi) \land (\psi \text{ nand } \psi)) \leftrightarrow \neg(\neg\varphi \land \neg\psi) \leftrightarrow \varphi \lor \psi
\]
\[
\varphi \text{ nor } \varphi \leftrightarrow \neg((\varphi \lor \varphi) \leftrightarrow \neg\varphi) \\
(\varphi \text{ nor } \varphi) \text{ nor } (\psi \text{ nor } \psi) \leftrightarrow \neg((\varphi \text{ nor } \varphi) \lor (\psi \text{ nor } \psi)) \leftrightarrow \neg(\neg\varphi \lor \neg\psi) \leftrightarrow \varphi \land \psi
\]