Def: Say a truth assignment $\nu$ satisfies $\varphi$ iff $\nu(\varphi) = T$.
(Note that all sentence symbols in the wff $\varphi$ must be in the domain of $\nu$.)

Def: Given a set $\Sigma$ of wffs (the hypotheses) and a wff $\psi$ (the conclusion).
Say that $\Sigma$ tautologically implies $\psi$ (written $\Sigma \models \psi$) if for every truth assignment $\nu$ (whose domain is the set of sentence symbols in $\Sigma$ and $\psi$) that satisfies every wff in $\Sigma$, then $\nu$ also satisfies $\psi$.

"Every time the hypotheses are true, then so is the conclusion."

Examples:
1. $\Sigma: A, (A \rightarrow B) \models B$
   Proof: There are for cases:
   - $\nu(A) = \nu(B) = T$
     \[ \begin{align*}
     \nu(A) &= T, \\
     \nu(B) &= T
     \end{align*} \]
     Then $\nu((A \rightarrow B)) = T$ \[ \begin{align*}
     \nu((A \rightarrow B)) &= T, \\
     \therefore \text{Hypotheses are satisfied.}
     \end{align*} \]
   - $\nu(A) = T, \nu(B) = F$
     \[ \begin{align*}
     \nu(A) &= T, \\
     \nu(B) &= F
     \end{align*} \]
     Then $\nu(A) = T$, $\nu((A \rightarrow B)) = F$ \[ \begin{align*}
     \nu(A) &= T, \\
     \nu(B) &= F \\
     \nu((A \rightarrow B)) &= F
     \end{align*} \]
     $\nu((A \rightarrow B)) = F$ \[ \begin{align*}
     \nu((A \rightarrow B)) &= F
     \end{align*} \]
     Hypotheses are not all satisfied.
   - $\nu(A) = F, \nu(B) = T$
     \[ \begin{align*}
     \nu(A) &= F, \\
     \nu(B) &= T
     \end{align*} \]
     Then $\nu(A) = F$ and $\nu((A \rightarrow B)) = T$ \[ \begin{align*}
     \nu((A \rightarrow B)) &= T \\
     \nu((A \rightarrow B)) &= T
     \end{align*} \]
     Conclusion is satisfied.
   - $\nu(A) = F$
     \[ \begin{align*}
     \nu(A) &= F
     \end{align*} \]
     Then $\nu(A) = F$ and $\nu((A \rightarrow B)) = F$ \[ \begin{align*}
     \nu((A \rightarrow B)) &= F
     \end{align*} \]
     Whenever the hypotheses are satisfied so is the conclusion.

\[ \square \]
2 \{A, (A \land A)\} \not\subseteq B

Proof: There is no \(u\) s.t. \(u(A) = T\) and \(u((A \land A)) = T\), so vacuously this holds. \(\Box\)

3 \(\emptyset \subseteq (A \supset A)\)

Proof: The empty set of hypotheses is always satisfied. If \(u(A) = T\) then \(\top ((A \supset A)) = T\). If \(u(A) = F\), then \(\top ((A \supset A)) = T\). \(\Box\)

Notation

\(\emptyset\) is the empty set.

- Write \(\top \phi\) in place of \(\emptyset \models \phi\).
- Write \(\square \phi\) in place of \((\square \top \phi)\).
- Write \(\top \phi\) in place of \((\top \phi \land \phi)\).

Def:

- Say that \(\sigma\) and \(\phi\) are **tautologically equivalent** iff \(\sigma \models \phi\).

- A wff \(\phi\) is a **tautology** iff \(\top \models \phi\).

- A wff \(\phi\) is **satisfiable** iff there exists a truth assignment \(u\) which satisfies \(\phi\).

Remark: If \(\phi\) is a tautology, then it is satisfiable. (Prove it!)

However not all wffs are tautologies, e.g., \((A \lor B)\). Not all wffs are satisfiable, e.g., \((A \supset (\neg A))\).

Test your understanding: Let \(\Delta \subseteq \Sigma\) be sets of wffs. Let \(\phi\) be a wff. Prove if \(\Delta \not\models \phi\) then \(\Sigma \not\models \phi\).

(Hint: Jabberwock.)
Truth tables

We can compute whether \( \sigma_1, ..., \sigma_n \models \varphi \) using a truth table.

Example \( \neg(A \lor B) \equiv (\neg A) \land (\neg B) \)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>( \neg(A \lor B) )</th>
<th>( (\neg A) \land (\neg B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</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>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Every time there is a T in the first column, there is a T in the second column.

Example \( \neg(A \land B) \equiv (\neg A) \lor (\neg B) \)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>( \neg(A \land B) )</th>
<th>( (\neg A) \lor (\neg B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Each column is the same.

Example \( \neg(A \land B) \not\equiv (\neg A) \land (\neg B) \)

true here, but false here
Example: \( F((A \lor (B \land C)) \iff (A \lor B) \land (A \lor C)) \)

Can be clever to avoid 8 truth assignments

<table>
<thead>
<tr>
<th>ABC</th>
<th>((A \lor (B \land C)) \iff (A \lor B) \land (A \lor C))</th>
</tr>
</thead>
<tbody>
<tr>
<td>T*F</td>
<td>TT (T) (T) (T) (T) (T) (T)</td>
</tr>
<tr>
<td>F*F</td>
<td>FF (T) (F) (F) (F) (F) (F)</td>
</tr>
<tr>
<td>F*F</td>
<td>FF (T) (F) (F) (F) (F) (F)</td>
</tr>
<tr>
<td>FTT</td>
<td>FTTTTT (T) (F) (T) (T) (F) (T)</td>
</tr>
</tbody>
</table>

Remarks on truth tables

- For \(n\) sentence symbols, need \(2^n\) rows. This is really big, even for a computer; \(2^{100}\) is more than the number of atoms in the universe.

- There are other methods for solving tautology, equivalence and satisfiability:
  1. Hilbert style axiom systems
  2. Natural deduction
  3. Sequent calculus
  4. Resolution theorem provers (for computers)

In practice, computers can solve large problems in sentential logic quickly. For example, a Sudoku puzzle can be expressed as a with with \(9 \times 9 \times 9\) sentence symbols is satisfiable (let, say, \(A_{1,2,3}\) mean that the square in row 1, column 2 is \(3\)).

In theory, these problems are difficult (specifically, satisfiability of sentential logic is a \(NP\)-complete problem).