10.1 Elements of the language

- Principle of Bivalence: each sentence is either true or false, never both, never neither.
- Each atomic sentence is a sentence.
- Syntax: if $\Phi$ and $\Psi$ are sentences, then the following are also sentences
  - $\neg \Phi$
  - $(\Phi \rightarrow \Psi)$
  - $(\Phi \land \Psi)$
  - $(\Phi \lor \Psi)$
  - $(\Phi \leftrightarrow \Psi)$
- Semantics: if $\Phi$ and $\Psi$ are sentences, then the meanings of the connectives are fully given by their truth tables. These truth tables are:

<table>
<thead>
<tr>
<th>$\Phi$</th>
<th>$\neg \Phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
\[
\begin{array}{c|c|c|c|c|c|c}
\Phi & \Psi & (\Phi \rightarrow \Psi) \\
\hline
T & T & T \\
T & F & F \\
F & T & T \\
F & F & T \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c}
\Phi & \Psi & (\Phi \land \Psi) \\
\hline
T & T & T \\
T & F & F \\
F & T & F \\
F & F & F \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c}
\Phi & \Psi & (\Phi \lor \Psi) \\
\hline
T & T & T \\
T & F & T \\
F & T & T \\
F & F & F \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c}
\Phi & \Psi & (\Phi \leftrightarrow \Psi) \\
\hline
T & T & T \\
T & F & F \\
F & T & F \\
F & F & T \\
\end{array}
\]
• A sentence of the propositional logic that must be true is a tautology.
• A sentence that must be false is a contradictory sentence.
• A sentence that is neither a tautology nor a contradictory sentence is a contingent sentence.
• Two sentences $\Phi$ and $\Psi$ are equivalent, or logically equivalent, when $(\Phi \rightarrow \Psi)$ is a theorem.

10.2 Reasoning with the language

• An argument is an ordered list of sentences, one sentence of which we call the “conclusion” and the others of which we call the “premises”.
• A valid argument is an argument in which: necessarily, if the premises are true, then the conclusion is true.
• A sound argument is a valid argument with true premises.
• Inference rules allow us to write down a sentence that must be true, assuming that certain other sentences are true. We say that the new sentence is “derived from” those other sentences using the inference rule.
• Schematically, we can write out the inference rules in the following way (think of these as saying, if you have written the sentence(s) above the line, then you can write the sentence below the line):

<table>
<thead>
<tr>
<th>Modus ponens</th>
<th>Modus tollens</th>
<th>Double negation</th>
<th>Double negation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\Phi \rightarrow \Psi)$</td>
<td>$(\Phi \rightarrow \Psi)$</td>
<td>$\Phi$</td>
<td>$\neg\neg\Phi$</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>$\neg\Psi$</td>
<td>$\Phi$</td>
<td>$\neg\Psi$</td>
</tr>
<tr>
<td>$\Psi$</td>
<td>$\neg\Phi$</td>
<td>$\neg\Phi$</td>
<td>$\Phi$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Addition</th>
<th>Addition</th>
<th>Modus tollendo ponens</th>
<th>Modus tollendo ponens</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Phi$</td>
<td>$\Psi$</td>
<td>$(\Phi \lor \Psi)$</td>
<td>$(\Phi \lor \Psi)$</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>$\Psi$</td>
<td>$\neg\Phi$</td>
<td>$\neg\Psi$</td>
</tr>
<tr>
<td>$(\Phi \lor \Psi)$</td>
<td>$(\Phi \lor \Psi)$</td>
<td>$\Psi$</td>
<td>$\Phi$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Adjunction</th>
<th>Simplification</th>
<th>Simplification</th>
<th>Bicondition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Phi$</td>
<td>$(\Phi \land \Psi)$</td>
<td>$(\Phi \land \Psi)$</td>
<td>$(\Phi \rightarrow \Psi)$</td>
</tr>
<tr>
<td>$\Psi$</td>
<td></td>
<td></td>
<td>$(\Psi \rightarrow \Phi)$</td>
</tr>
</tbody>
</table>
• A proof (or derivation) is a syntactic method for showing an argument is valid. Our system has three kinds of proof (or derivation): direct, conditional, and indirect.

• A direct proof (or direct derivation) is an ordered list of sentences in which every sentence is either a premise or is derived from earlier lines using an inference rule. The last line of the proof is the conclusion.

• A conditional proof (or conditional derivation) is an ordered list of sentences in which every sentence is either a premise, is the special assumption for conditional derivation, or is derived from earlier lines using an inference rule. If the assumption for conditional derivation is \( \Phi \), and we derive as some step in the proof \( \Psi \), then we can write after this \( (\Phi \rightarrow \Psi) \) as our conclusion.

• An indirect proof (or indirect derivation, and also known as a reductio ad absurdum) is: an ordered list of sentences in which every sentence is either 1) a premise, 2) the special assumption for indirect derivation (also sometimes called the “assumption for reductio”), or 3) derived from earlier lines using an inference rule. If our assumption for indirect derivation is \( \neg \Phi \), and we derive as some step in the proof \( \Psi \) and also as some step of our proof \( \neg \Psi \), then we conclude that \( \Phi \).

• We can use Fitch bars to write out the three proof schemas in the following way:
A sentence that we can prove without premises is a theorem.

Suppose $\Phi$ is a theorem, and it contains the atomic sentences $P_1 \ldots P_n$. If we replace each and every occurrence of one of those atomic sentences $P_i$ in $\Phi$ with another sentence $\Psi$, the resulting sentence is also a theorem. This can be repeated for any atomic sentences in the theorem.