Propositional logic

- **Proposition**: A statement which can be true or false.
- **Coq**: \( P : \text{Prop} \) means \( P \) is a proposition.
- **Propositional variables**: stand for any proposition; e.g. in Coq:
  
  Variables \( P \ Q \ R : \text{Prop} \).

- **Tautology**: A proposition containing propositional variables which is always true.

- **Propositional constants** \( \text{True} \), \( \text{False} \).

- **Basic propositional connectives**:

<table>
<thead>
<tr>
<th>Name</th>
<th>Math</th>
<th>Coq</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implication</td>
<td>( P \to Q )</td>
<td>( P \rightarrow Q )</td>
<td>If ( P ) then ( Q )</td>
</tr>
<tr>
<td>Conjunction</td>
<td>( P \land Q )</td>
<td>( P \land Q )</td>
<td>( P ) and ( Q )</td>
</tr>
<tr>
<td>Disjunction</td>
<td>( P \lor Q )</td>
<td>( P \lor Q )</td>
<td>( P ) or ( Q )</td>
</tr>
<tr>
<td>Equivalence</td>
<td>( P \leftrightarrow Q )</td>
<td>( P \leftrightarrow Q )</td>
<td>( P ) if and only if ( Q )</td>
</tr>
<tr>
<td>Negation</td>
<td>( \neg P )</td>
<td>( \neg P )</td>
<td>not ( P )</td>
</tr>
</tbody>
</table>
→ is right-associative, i.e.

\[ P \rightarrow Q \rightarrow R = P \rightarrow (Q \rightarrow R) \]

\( \lor, \land \) bind stronger than \( \rightarrow \) (and \( \leftrightarrow \)), i.e.

\[ P \lor Q \rightarrow R = (P \lor Q) \rightarrow R \]

\( \land \) binds stronger than \( \lor \):

\[ P \lor Q \land R = P \lor (Q \land R) \]

\( \neg \) binds stronger than \( \land \):

\[ \neg P \land Q = (\neg P) \land Q \]
Start a proof with \texttt{Lemma} or \texttt{Theorem}. Give it a name! E.g. \texttt{Lemma andCom : P \land Q \rightarrow Q \land P}.

Coq displays the \textit{proof state}. The user issues \textit{tactics} until Coq says \texttt{Proof completed}.

Finish with \texttt{Qed}.
Leaves the proof state and saves the proof under the given name.

Read \texttt{p : P} as \texttt{p} is a proof of \texttt{P}, e.g. \texttt{andCom : P \land Q \rightarrow Q \land P}.
Coq proof state (example)

2 subgoals
H : P \(\land\) Q
H1 : P
H2 : Q

============================
Q

subgoal 2 is:
P

- Two subgoals: currently proving \(Q\), when we are finished we prove \(P\).
- Assumptions above ===...===.
- Assumptions have names, e.g. \(H, H1, H2\)
- Current goal (e.g. \(Q\)) is below the line.
- If current Goal = one of the assumptions, use \texttt{exact}, e.g. \texttt{exact H2}.
Proof rules / basic tactics

- General pattern

  \[\text{premise (what we need to show)} \quad \frac{}{\text{name of the rule}} \quad \text{conclusion (what we want to show)}\]

- Read proof rules from bottom to top!
- We write \( \Gamma \vdash P \) for
  
  \textit{From the set of assumptions } \Gamma \text{ (Gamma), we can prove } P.
- The symbol \( \vdash \) (turnstile) replaces Coq’s \( \equiv \ldots \equiv \)
- Example:

  \[
  H : P \in \Gamma \\
  \frac{}{\Gamma \vdash P} \quad \text{exact } H
  \]

- Read \( H : P \in \Gamma \) as \( H : P \) occurs in \( \Gamma \).
Rules for implication

\[ \Gamma, H : P \vdash Q \]
\[ \Gamma \vdash P \rightarrow Q \]

**intro**: to prove \( P \rightarrow Q \), assume \( P \) and prove \( Q \).

**apply**: If we know \( P \rightarrow Q \) then to prove \( Q \) it is enough to prove \( P \).

The actual behaviour of **apply** is more subtle!

See examples in l01.v
Rules for conjunction

\[ \Gamma \vdash P \quad \Gamma \vdash Q \]
\[ \frac{}{\Gamma \vdash P \land Q} \text{split} \]

\[ \Gamma \vdash P \rightarrow Q \rightarrow R \]
\[ \frac{H : P \land Q \in \Gamma}{\Gamma \vdash R} \text{elim}_H \]

- **split**: to prove \( P \land Q \) prove \( P \) and then \( Q \).
- **elim**: If we know \( P \land Q \) then to prove \( R \) it is enough to prove \( P \rightarrow Q \rightarrow R \).
- See examples in l02.v
Rules for disjunction

\[
\frac{\Gamma \vdash P}{\Gamma \vdash P \lor Q} \text{ left } \quad \frac{\Gamma \vdash Q}{\Gamma \vdash P \lor Q} \text{ right }
\]

\[
\frac{H : P \lor Q \in \Gamma}{\Gamma \vdash P \rightarrow R \quad \Gamma \vdash Q \rightarrow R \quad \Gamma \vdash R} \text{ case H}
\]

- **left**: to prove \( P \lor Q \) prove \( P \).
- **right**: to prove \( P \lor Q \) prove \( Q \).
- **case**: If we know \( P \lor Q \) then to prove \( R \) it is enough to prove \( P \rightarrow R \) and \( Q \rightarrow R \).
- See examples in l02.v
Rules for True and False

\[ \Gamma \vdash \text{True} \quad \text{split} \]
\[ \Gamma \vdash R \quad \text{case } H \]

- **split**: to prove True you need to prove nothing.
- **case**: if you know False you can prove anything.
### Summary

<table>
<thead>
<tr>
<th>connective</th>
<th>Introduction</th>
<th>Elimination</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P \rightarrow Q$</td>
<td>intro(s)</td>
<td>apply $Hyp$</td>
</tr>
<tr>
<td>$P \land Q$</td>
<td>split</td>
<td>elim $Hyp$</td>
</tr>
<tr>
<td>True</td>
<td>split</td>
<td></td>
</tr>
<tr>
<td>$P \lor Q$</td>
<td>left, right</td>
<td>case $Hyp$</td>
</tr>
<tr>
<td>False</td>
<td></td>
<td>case $Hyp$</td>
</tr>
</tbody>
</table>
Defined connectives

negation

\[ \neg P = P \rightarrow \text{False} \]

iff

\[ P \leftrightarrow Q = (P \rightarrow Q) \land (Q \rightarrow P) \]