# Logik für Informatiker
# Logic for computer scientists

Till Mossakowski

WiSe 2013/14

# Proofs for Boolean Logic

**Conjunction Elimination ($\wedge$ Elim)**

$$
\begin{array}{l|l}
 & P_1 \wedge \ldots \wedge P_i \wedge \ldots \wedge P_n \\
 & \vdots \\
\triangleright & P_i
\end{array}
$$

## Conjunction Introduction (∧ Intro)

$$\begin{array}{l|l} & P_1 \\ & \Downarrow \\ & P_n \\ & \vdots \\ \triangleright & P_1 \wedge \ldots \wedge P_n \end{array}$$

**Disjunction Introduction**
**(∨ Intro)**

$$
\begin{array}{l|l}
 & P_i \\
 & \vdots \\
\triangleright & P_1 \lor \ldots \lor P_i \lor \ldots \lor P_n
\end{array}
$$

## Disjunction Elimination ($\vee$ Elim)

$P_1 \vee \ldots \vee P_n$

$\vdots$

$P_1$

$\vdots$

S

$\Downarrow$

$P_n$

$\vdots$

S

$\vdots$

$\triangleright$ S

## $\perp$ Introduction ($\perp$ Intro)

$$
\begin{array}{l|l}
 & P \\
 & \vdots \\
 & \neg P \\
 & \vdots \\
\triangleright & \perp
\end{array}
$$

## Negation Introduction (¬ Intro)

$$
\begin{array}{l|l}
 & \quad \begin{array}{|l} P \\ \hline \vdots \\ \bot \end{array} \\
\triangleright & \neg P
\end{array}
$$

**Negation Elimination (¬ Elim)**

$$\begin{array}{l|l} & \neg\neg P \\ & \vdots \\ \triangleright & P \end{array}$$

**$\perp$ Elimination ($\perp$ Elim)**

$$\begin{array}{l|l} & \perp \\ & \vdots \\ \triangleright & P \end{array}$$

# Strategies and tactics in Fitch

1. Understand what the sentences are saying.
2. Decide whether you think the conclusion follows from the premises.
3. If you think it does not follow, or are not sure, try to find a counterexample.
4. If you think it does follow, try to give an informal proof.
5. If a formal proof is called for, use the informal proof to guide you in finding one.
6. In giving consequence proofs, both formal and informal, don't forget the tactic of working backwards.
7. In working backwards, though, always check that your intermediate goals are consequences of the available information.

# Strategies in Fitch, cont'd

- Always try to match the situation in your proof with the rules in the book (see book appendix for a complete list)
- Look at the main connective in a premise, apply the corresponding elimination rule (forwards)
- Or: look at the main connective in the conclusion, apply the corresponding introduction rule (backwards)

# Conditionals

# Conditionals

| P | Q | P → Q |
|---|---|---|
| T | T | **T** |
| T | F | **F** |
| F | T | **T** |
| F | F | **T** |

Game rule: $P \rightarrow Q$ is replaced by $\neg P \vee Q$.

# Formalisation of conditional sentences

- The following English constructions are all translated $P \rightarrow Q$: If $P$ then $Q$; $Q$ if $P$; $P$ only if $Q$; and Provided $P$, $Q$.
- Unless $P$, $Q$ and $Q$ unless $P$ are translated: $\neg P \rightarrow Q$.
- $Q$ is a logical consequence of $P_1, \ldots, P_n$ if and only if the sentence $(P1 \wedge \cdots \wedge P_n) \rightarrow Q$ is a logical truth.

**Conditional Elimination ($\to$ Elim)**

$$
\begin{array}{l|l}
 & P \to Q \\
 & \vdots \\
 & P \\
 & \vdots \\
\triangleright & Q
\end{array}
$$

### Conditional Introduction ($\rightarrow$ Intro)

$$
\begin{array}{l|l}
 & \quad \begin{array}{|l} P \\ \hline \vdots \\ Q \end{array} \\
\triangleright & P \rightarrow Q
\end{array}
$$

# Biconditionals

| P | Q | $P \leftrightarrow Q$ |
|---|---|---|
| T | T | **T** |
| T | F | **F** |
| F | T | **F** |
| F | F | **T** |

Game rule: $P \leftrightarrow Q$ is replaced by $(P \rightarrow Q) \wedge (Q \rightarrow P)$.

# Biconditionals and logical equivalence

$P$ and $Q$ are logically equivalent ($P \Leftrightarrow Q$)
if and only if
the sentence $P \leftrightarrow Q$ is a logical truth.

Note that $P \Leftrightarrow Q$ is a meta statement, not a sentence of PL1.

# Conversational implicature

"Max is home unless Claire is at the library" can be formalised as

$$\neg Library(claire) \rightarrow Home(max)$$

but many people would formalise it as

$$\neg Library(claire) \leftrightarrow Home(max)$$

The part

$$\neg Library(claire) \leftarrow Home(max)$$

is called conservational implicature. It is possibly, but not necessarily meant by the English sentence.

An addition can cancel the implicature:
"On the other hand, if Claire is at the library, I have no idea where Max is."

# Truth-functional completeness

## Definition

A logical connective is truth-functional, if the truth value of a complex sentence built up using these connectives depends on nothing more than the truth values of the simpler sentences from which it is built.

Truth-functional: $\wedge$, $\vee$, $\neg$, $\leftarrow$, $\leftrightarrow$

Not truth-functional: because, after, necessarily

## Definition

A set of logical connectives is truth-functionally complete, if it suffices to express every truth-functional connective.

## Theorem

*The set* $\{\wedge, \vee, \neg\}$ *is truth-functionally complete.*

# Example: a binary truth-functional connective

| P | Q | Neither P nor Q | |
|---|---|---|---|
| T | T | F | |
| T | F | F | |
| F | T | F | |
| F | F | T | $\neg P \wedge \neg Q$ |

Neither P nor Q can be expressed as $\neg P \wedge \neg Q$.

# Example: a ternary truth-functional connective

| P | Q | R | ♣(P,Q,R) | |
|---|---|---|---|---|
| T | T | T | T | $P \wedge Q \wedge R$ |
| T | T | F | T | $P \wedge Q \wedge \neg R$ |
| T | F | T | F | |
| T | F | F | F | |
| F | T | T | T | $\neg P \wedge Q \wedge R$ |
| F | T | F | F | |
| F | F | T | T | $\neg P \wedge \neg Q \wedge R$ |
| F | F | F | F | |

♣(P,Q,R) can be expressed as

$$(P \wedge Q \wedge R) \vee (P \wedge Q \wedge \neg R) \vee (\neg P \wedge Q \wedge R) \vee (\neg P \wedge \neg Q \wedge R).$$

# Logical equivalences of (bi)conditionals

$$
\begin{array}{rcl}
P \to Q & \Leftrightarrow & \neg Q \to \neg P \\
P \to Q & \Leftrightarrow & \neg P \vee Q \\
\neg(P \to Q) & \Leftrightarrow & P \wedge \neg Q \\
P \leftrightarrow Q & \Leftrightarrow & (P \to Q) \wedge (Q \to P) \\
P \leftrightarrow Q & \Leftrightarrow & (P \wedge Q) \vee (\neg P \wedge \neg Q)
\end{array}
$$

**Biconditional Elimination ($\leftrightarrow$ Elim)**

$$
\begin{array}{l|l}
 & \mathsf{P} \leftrightarrow \mathsf{Q} \text{ (or } \mathsf{Q} \leftrightarrow \mathsf{P}) \\
 & \vdots \\
 & \mathsf{P} \\
 & \vdots \\
\triangleright & \mathsf{Q}
\end{array}
$$

## Biconditional Introduction ($\leftrightarrow$ Intro)

$$
\begin{array}{l|l}
 & \begin{array}{|l} P \\ \hline \vdots \\ Q \end{array} \\
 & \begin{array}{|l} Q \\ \hline \vdots \\ P \end{array} \\
\rhd & P \leftrightarrow Q
\end{array}
$$

**Reiteration (Reit)**

$$\begin{array}{l|l} & P \\ & \vdots \\ \triangleright & P \end{array}$$

# Soundness and completeness

# Object and meta theory

Object theory = reasoning within a formal proof system (e.g. Fitch)

Meta theory = reasoning about a formal proof system

# Tautological consequence

A sentence $S$ is a tautological consequence of a set of sentences $\mathcal{T}$, written

$$\mathcal{T} \models_T S,$$

if all valuations of atomic formulas with truth values that make all sentences in $\mathcal{T}$ true also make $S$ true.

# Propositional proofs

$S$ is $\mathcal{F}_T$-provable from $\mathcal{T}$, written

$$\mathcal{T} \vdash_T S,$$

if there is a formal proof of $S$ with premises drawn from $\mathcal{T}$ using the elimination and introduction rules for $\vee, \wedge, \neg, \rightarrow, \leftrightarrow$ and $\bot$.

Again note: $\mathcal{T}$ may be infinite.

# Soundness

Theorem 1. The proof calculus $\mathcal{F}_T$ is sound, i.e. if

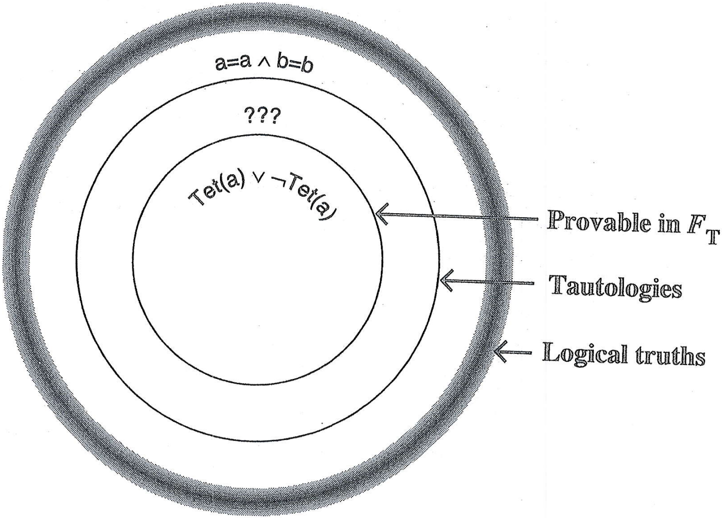
$$\mathcal{T} \vdash_T S,$$

then

$$\mathcal{T} \models_T S.$$

Proof: Book: by contradiction, using the first invalid step.
Here: by induction on the length of the proof.

# Soundness



# Completeness

**Theorem 2** (Bernays, Post). The proof calculus $\mathcal{F}_T$ is complete, i.e. if

$$\mathcal{T} \models_T S,$$

then

$$\mathcal{T} \vdash_T S.$$

This theorem will be proved later in the lecture.

# Completeness