Propositional Logic – Language

- A logic consists of:
  - an alphabet $\mathcal{A}$,
  - a language $\mathcal{L}$, i.e., a set of formulas, and
  - a binary relation $\models$ between a set of formulas and a formula.
- An alphabet $\mathcal{A}$ consists of
  - a finite or countably infinite set of nullary relation symbols $\mathcal{A}_R = \{p, q, \ldots\}$,
  - the set of connectives $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ and
  - the punctuation symbols "(" and ")".
- The language $\mathcal{L}$ of propositional logic formulas is defined as follows:
  - Each relation symbol is a formula.
  - If F and G are formulas, then $\neg F$, $(F \wedge G)$, $(F \vee G)$, $(F \rightarrow G)$ and $(F \leftrightarrow G)$ are formulas.

Notions, Remarks and an Example

- Nullary relation symbols are also called propositional variables, nullary predicate symbols and atoms.
- Literals are formulas of the form $\boldsymbol{p}$ or $\neg \boldsymbol{p}$.
- We define a precedence hierarchy as follows:

$$\neg > \wedge > \vee > \{\leftarrow, \rightarrow\} > \leftrightarrow .$$

- A simple example: $(\boldsymbol{p}_1 \vee \boldsymbol{p}_2) \wedge (\boldsymbol{q}_1 \vee \boldsymbol{q}_2)$.

Structural Induction and Recursion

- Theorem 2.1 (Principle of Structural Induction) Every formula of propositional logic has a certain property $\boldsymbol{E}$ provided that:

    1. Basis step: Every propositional variable has property $\boldsymbol{E}$.
    2. Induction steps:
       If $\boldsymbol{F}$ has property $\boldsymbol{E}$ so does $\neg\boldsymbol{F}$.
       If $\boldsymbol{F}$ and $\boldsymbol{G}$ have property $\boldsymbol{E}$ so does $\boldsymbol{F} \circ \boldsymbol{G}$,
       where $\circ \in \{\wedge, \vee, \leftarrow, \rightarrow, \leftrightarrow\}$.

- Theorem 2.2 (Principle of Structural Recursion) There is one, and only one, function $\boldsymbol{h}$ defined on the set of propositional formulas such that:

    1. Basis step: The value of $\boldsymbol{h}$ is specified explicitly on propositional variables.
    2. Recursion steps:
       The value of $\boldsymbol{h}$ on $\neg\boldsymbol{F}$ is specified in terms of the value of $\boldsymbol{h}$ on $\boldsymbol{F}$.
       The value of $\boldsymbol{h}$ on $\boldsymbol{F} \circ \boldsymbol{G}$ is specified in terms of the values of $\boldsymbol{h}$ on $\boldsymbol{F}$ and on $\boldsymbol{G}$,
       where $\circ \in \{\wedge, \vee, \leftarrow, \rightarrow, \leftrightarrow\}$.

Subformulas

- The set of subformulas $\mathcal{T}(H)$ is the smallest set satisfying the following conditions:
  - $H \in \mathcal{T}(H)$.
  - If $\neg F \in \mathcal{T}(H)$, then $F \in \mathcal{T}(H)$.
  - If $F \circ G \in \mathcal{T}(H)$, then $F, G \in \mathcal{T}(H)$,
    where $\circ \in \{\wedge, \vee, \leftarrow, \rightarrow, \leftrightarrow\}$.

Propositional Logic – Semantics

- Goal: assign a meaning to formulas with the help of a function $\mathcal{L} \longrightarrow \{t, f\}$.
- Interpretation I: a total mapping $\mathcal{A}_R \longrightarrow \{t,\ f\}$.
- Abbreviation: $I \subset \mathcal{A}_R$ with the understanding that $I(p) = t$ iff $p \in I$.
- I models F, in symbols $I \models F$:

$$
\begin{array}{lll}
I \models p & \text{iff} & I(p) = t \ \ (\text{iff } p \in I) \\
I \models \neg F & \text{iff} & I \not\models F \\
I \models F_1 \wedge F_2 & \text{iff} & I \models F_1 \text{ and } I \models F_2 \\
I \models F_1 \vee F_2 & \text{iff} & I \models F_1 \text{ or } I \models F_2 \\
I \models F_1 \longrightarrow F_2 & \text{iff} & I \not\models F_1 \text{ or } I \models F_2 \\
I \models F_1 \leftrightarrow F_2 & \text{iff} & I \models F_1 \longrightarrow F_2 \text{ and } I \models F_2 \longrightarrow F_1.
\end{array}
$$

Connectives

- Unary connective:

	$\neg$
$\boldsymbol{t}$	$\boldsymbol{f}$
$\boldsymbol{f}$	$\boldsymbol{t}$

- Binary connectives:

		$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$
$\boldsymbol{t}$	$\boldsymbol{t}$	$\boldsymbol{t}$	$\boldsymbol{t}$	$\boldsymbol{t}$	$\boldsymbol{t}$
$\boldsymbol{t}$	$\boldsymbol{f}$	$\boldsymbol{f}$	$\boldsymbol{t}$	$\boldsymbol{f}$	$\boldsymbol{f}$
$\boldsymbol{f}$	$\boldsymbol{t}$	$\boldsymbol{f}$	$\boldsymbol{t}$	$\boldsymbol{t}$	$\boldsymbol{f}$
$\boldsymbol{f}$	$\boldsymbol{f}$	$\boldsymbol{f}$	$\boldsymbol{f}$	$\boldsymbol{t}$	$\boldsymbol{t}$

Models, Validity and More Notions

- An interpretation I for F is said to be a model for F iff $I \models F$.
- Let $\mathcal{F}$ be a set of formulas.
  I is a model for $\mathcal{F}$ iff I is a model for each $G \in \mathcal{F}$.
- F is said to be valid or a tautology iff for all I we find that $I \models F$.
- F is said to be satisfiable iff there exists an I such that $I \models F$.
- F is said to be falsifiable iff there exists an I such that $I \not\models F$.
- F is said to be unsatisfiable iff for all I we find that $I \not\models F$.

Logical Entailment

- A set of formulas $\mathcal{F}$ logically entails G (or G is a logical consequence of $\mathcal{F}$ or G is a theorem of $\mathcal{F}$), in symbols $\mathcal{F} \models G$, iff each model for $\mathcal{F}$ is also a model for G.
- A satisfiable set of formulas together with all its theorems is said to be a theory.
- Truth tabling: $\{p \rightarrow q \wedge r\} \models p \rightarrow r$?

p	q	r	$q \wedge r$	$p \rightarrow q \wedge r$	$p \rightarrow r$
t	t	t	t	t	t
t	t	f	f	f	f
t	f	t	f	f	t
t	f	f	f	f	f
f	t	t	t	t	t
f	t	f	f	t	t
f	f	t	f	t	t
f	f	f	f	t	t

Logical Consequence vs. Validity vs. Unsatisfiability

- Theorem 2.3: $\boldsymbol{\mathcal{F} \cup \{G\} \models H}$ iff $\boldsymbol{\mathcal{F} \models G \rightarrow H}$.
  - $\{F_1, \ldots, F_n\} \models F$
    iff $\{F_1, \ldots, F_{n-1}\} \models F_n \rightarrow F$
    iff $\ldots$
    iff $\emptyset \models F_1 \rightarrow (F_2 \ldots \rightarrow (F_n \rightarrow F) \ldots)$.
  - Logical consequence is related to validity.

- Abbreviation: $\emptyset \models F \;\rightsquigarrow\; \models F$.

- Theorem 2.4: F is valid iff $\neg F$ is unsatisfiable.
  - Validity is related unsatisfiability.
  - Logical consequence is related to unsatisfiability

Semantic Equivalence

- $F \equiv G$ if for all interpretations I we find that $I \models F$ iff $I \models G$.
- Some equivalences:

$$
\begin{array}{rcll}
(F \wedge F) & \equiv & F & \\
(F \vee F) & \equiv & F & \text{idempotency} \\
(F \wedge G) & \equiv & (G \wedge F) & \\
(F \vee G) & \equiv & (G \vee F) & \text{commutativity} \\
((F \wedge G) \wedge H) & \equiv & (F \wedge (G \wedge H)) & \\
((F \vee G) \vee H) & \equiv & (F \vee (G \vee H)) & \text{associativity} \\
((F \wedge G) \vee F) & \equiv & F & \\
((F \vee G) \wedge F) & \equiv & F & \text{absorption} \\
(F \wedge (G \vee H)) & \equiv & ((F \wedge G) \vee (F \wedge H)) & \\
(F \vee (G \wedge H)) & \equiv & ((F \vee G) \wedge (F \vee H)) & \text{distributivity}
\end{array}
$$

More Equivalences

$$\neg\neg F \equiv F \qquad \text{double negation}$$

$$\begin{aligned} \neg(F \wedge G) &\equiv (\neg F \vee \neg G) \\ \neg(F \vee G) &\equiv (\neg F \wedge \neg G) \end{aligned} \qquad \text{de Morgan}$$

$$\begin{aligned} (F \vee G) &\equiv F, \text{ if } F \text{ tautology} \\ (F \wedge G) &\equiv G, \text{ if } F \text{ tautology} \end{aligned} \qquad \text{tautology}$$

$$\begin{aligned} (F \vee G) &\equiv G, \text{ if } F \text{ unsatisfiable} \\ (F \wedge G) &\equiv F, \text{ if } F \text{ unsatisfiable} \end{aligned} \qquad \text{unsatisfiability}$$

$$(F \leftrightarrow G) \equiv (F \to G) \wedge (G \to F) \qquad \text{equivalence}$$

$$(F \to G) \equiv (\neg F \vee G) \qquad \text{implication}$$

The Replacement Theorem

- $\boldsymbol{F}[\boldsymbol{G}]$: formula $\boldsymbol{F}$, in which the occurrences of the formula $\boldsymbol{G}$ are important.
- $\boldsymbol{F}[\boldsymbol{G}/\boldsymbol{H}]$: formula obtained from $\boldsymbol{F}$ by replacing all occurrences of $\boldsymbol{G}$ by $\boldsymbol{H}$.

- Theorem 2.5: If $\boldsymbol{G} \equiv \boldsymbol{H}$, then $\boldsymbol{F}[\boldsymbol{G}] \equiv \boldsymbol{F}[\boldsymbol{G}/\boldsymbol{H}]$.

Normal Forms

- Negation normal form: the formula is built solely by literals, conjunctions and disjunctions.
- Conjunctive normal form: the formula has the form $F_1 \wedge \ldots \wedge F_n$, $n \geq 0$, where each of $F_1, \ldots, F_n$ is a disjunction of literals.
  - case $n = 0$: $\langle\,\rangle$ denoting a valid formula.
- Disjunctive normal form: the formula has the form $F_1 \vee \ldots \vee F_n$, $n \geq 0$, where each of $F_1, \ldots, F_n$ is a conjunction of literals.
  - case $n = 0$: $[\,]$ denoting an unsatisfiable formula.

- Clause form: set notation of formulas in conjunctive normal form;
- Clauses: elements of these sets.
- Dual clause form: set notation of formulas in disjunctive normal form;
- Dual clauses: elements of these sets.

Normal Form Transformation

Input A propositional logic formula $\boldsymbol{F}$.

Output A propositional logic formula $\boldsymbol{G}$ in conjunctive normal form which is equivalent to $\boldsymbol{F}$.

1. Eliminate all equivalence signs using the equivalence law.
2. Eliminate all implication signs using the implication law.
3. Eliminate all negation signs except those in literals using the de Morgan and the double negation laws.
4. Distribute all disjunctions over conjunctions using the second distributivity, the commutativity and the associativity laws.

Normal Form Transformation - Example

$$
\begin{array}{rcll}
(p \wedge (q \rightarrow r)) \rightarrow s & \equiv & (p \wedge (\neg q \vee r)) \rightarrow s & \text{by implication} \\
& \equiv & \neg(p \wedge (\neg q \vee r)) \vee s & \text{by implication} \\
& \equiv & (\neg p \vee \neg(\neg q \vee r)) \vee s & \text{by de Morgan} \\
& \equiv & (\neg p \vee (\neg\neg q \wedge \neg r)) \vee s & \text{by de Morgan} \\
& \equiv & (\neg p \vee (q \wedge \neg r)) \vee s & \text{by double negation} \\
& \equiv & ((\neg p \vee q) \wedge (\neg p \vee \neg r)) \vee s & \text{by distributivity} \\
& \equiv & s \vee ((\neg p \vee q) \wedge (\neg p \vee \neg r)) & \text{by commutativity} \\
& \equiv & (s \vee (\neg p \vee q)) \wedge (s \vee (\neg p \vee \neg r) & \text{by distributivity} \\
& \equiv & (s \vee \neg p \vee q) \wedge (s \vee \neg p \vee \neg r) & \text{by associativity}
\end{array}
$$

Clause Form

- Conjunctive normal form: $(L_{11} \vee \ldots \vee L_{1n_1}) \wedge \ldots \wedge (L_{m1} \vee \ldots \vee L_{mn_m})$,
  Clause form: $\{\{L_{11}, \ldots, L_{1n_1}\}\ , \ldots ,\ \{L_{m1}, \ldots, L_{mn_m}\}\}$.
- Unit clause: clause which contains only one literal.
- Horn clause: clause which contains at most one positive literal
  $(L \leftarrow L_1 \wedge \ldots \wedge L_n)$.
- Goal clause: Horn clause which contains only negative literals
  $(\leftarrow L_1 \wedge \ldots \wedge L_n)$.
- Definite clause: Horn clause which contains a positive literal.
- Fact: definite unit clause.
- [ ]: denotes the empty clause.

Propositional Logic Programs

- Procedural vs. declarative reading of definite clauses.
- Definite propositional logic program $\mathcal{F}$: set of definite clauses.
- Proposition 2.1: If I_1 and I_2 are models of $\mathcal{F}$ then so is $I_1 \cap I_2$.
  - There exists a least model $M_{\mathcal{F}}$ of $\mathcal{F}$.
- Theorem 2.6: Let $\mathcal{F}$ be a logic program. $M_{\mathcal{F}} = \{p \mid \mathcal{F} \models p\}$.
- Meaning function:

$$T_{\mathcal{F}}(I) = \{p \mid p \leftarrow p_1 \wedge \ldots \wedge p_n \in \mathcal{F} \wedge \{p_1, \ldots, p_n\} \subseteq I\}.$$

$$\begin{array}{lcll} T_{\mathcal{F}}\uparrow\ 0 & = & \emptyset & \\ T_{\mathcal{F}}\uparrow\ (n+1) & = & T_{\mathcal{F}}(T_{\mathcal{F}}\uparrow\ n) & \text{for all } n \geq 0. \end{array}$$

- $T_{\mathcal{F}}$ admits a least fixed point lfp$(T_{\mathcal{F}})$.
- Theorem 2.7: $M_{\mathcal{F}} = \mathit{lfp}(T_{\mathcal{F}}) = \mathit{lub}(\{T_{\mathcal{F}} \uparrow\ n \mid n \in N\})$.

Calculus

- Logic: $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{L}}, \models \rangle$.
- Calculus: $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\Gamma}, \boldsymbol{\Pi} \rangle$, where $\boldsymbol{\Gamma}$ is a set of formulas called axioms and
  $\boldsymbol{\Pi}$ is a set of inference rules.
- $\boldsymbol{\mathcal{F}} \vdash \boldsymbol{F}$: inference relation defined by $\boldsymbol{\Gamma}$ and $\boldsymbol{\Pi}$.
- Soundness: If $\boldsymbol{\mathcal{F}} \vdash \boldsymbol{F}$ then $\boldsymbol{\mathcal{F}} \models \boldsymbol{F}$.
- Completeness: If $\boldsymbol{\mathcal{F}} \models \boldsymbol{F}$ then $\boldsymbol{\mathcal{F}} \vdash \boldsymbol{F}$

Classification of Calculi

A calculus is said to be

- negative if its axioms are unsatisfiable.
- positive if its axioms are valid,
- generating if theorems are derived from the axioms using the inference rules,
- analyzing if theorems are reduced to the axioms using the inference rules.

Natural Deduction

- Gentzen 1935: How to represent logical reasoning in mathematics?
  - Calculus of natural deduction.
- Alphabet: propositional logic.
  - $[\,]$ denotes some unsatisfiable formula.
  - $\langle\,\rangle$ denotes some valid formula.
- Language: propositional logic formulas.
- Axioms: $\{\langle\,\rangle\}$.

Natural Deduction - Inference Rules

- Constant Rules:

$$\frac{[\,]}{F}$$

- Negation Rules:

$$\frac{\begin{array}{c}F\\ \neg F\end{array}}{[\,]} \qquad \frac{\boxed{\begin{array}{c}F\\ \vdots\\ [\,]\end{array}}}{\neg F} \qquad \frac{\boxed{\begin{array}{c}\neg F\\ \vdots\\ [\,]\end{array}}}{F}$$

- Introduction Rules for Binary Connectives:

$\land I$

$$\frac{\begin{array}{c}F\\ G\end{array}}{F \land G}$$

$\lor I$

$$\frac{\boxed{\begin{array}{c}\neg F\\ \vdots\\ G\end{array}}}{F \lor G} \qquad \frac{\boxed{\begin{array}{c}\neg G\\ \vdots\\ F\end{array}}}{F \lor G}$$

$\longrightarrow I$

$$\frac{\boxed{\begin{array}{c}F\\ \vdots\\ G\end{array}}}{F \longrightarrow G} \qquad \frac{\boxed{\begin{array}{c}\neg G\\ \vdots\\ \neg F\end{array}}}{F \longrightarrow G}$$

- Elimination Rules for Binary Connectives:

$\land E$

$$\frac{F \land G}{F} \qquad \frac{F \land G}{G}$$

$\lor E$

$$\frac{\begin{array}{c}\neg F\\ F \lor G\end{array}}{G} \qquad \frac{\begin{array}{c}\neg G\\ F \lor G\end{array}}{F}$$

$\longrightarrow E$

$$\frac{\begin{array}{c}F\\ F \longrightarrow G\end{array}}{G} \qquad \frac{\begin{array}{c}\neg G\\ F \longrightarrow G\end{array}}{\neg F}$$

Deductions and Proofs

- A deduction in the calculus of natural deduction is a sequence of formulas possibly enclosed in (open or closed) boxes such that each element is either
  - the axiom $\langle\,\rangle$,
  - follows from earlier elements occurring in open boxes at this stage by one of the rules of inference or
  - is a formula different from $\langle\,\rangle$ and does not follow by one of the rules of inference, in which case the formula is called assumption and a new box is opened.

- A proof of $\boldsymbol{F}$ in the calculus of natural deduction is a deduction in which all boxes are closed and $\boldsymbol{F}$ is the last element in the deduction.

Lemmas

- The lemmas:

$$\frac{\neg\neg F}{F} \qquad \frac{\begin{array}{c}\neg(F \wedge G)\\ F\end{array}}{\neg G} \qquad \frac{\begin{array}{c}\neg(F \wedge G)\\ G\end{array}}{\neg F} \qquad \frac{\neg(F \vee G)}{(\neg F \wedge \neg G)}$$

- represent the sequence of formulas:

$$\begin{array}{c}\neg\neg F\\ \boxed{\begin{array}{c}\neg F\\ [\,]\end{array}}\\ F\end{array} \qquad \begin{array}{c}\neg(F \wedge G)\\ F\\ \boxed{\begin{array}{c}G\\ (F \wedge G)\\ [\,]\end{array}}\\ \neg G\end{array} \qquad \begin{array}{c}\neg(F \wedge G)\\ G\\ \boxed{\begin{array}{c}F\\ (F \wedge G)\\ [\,]\end{array}}\\ \neg F\end{array} \qquad \begin{array}{c}\neg(F \vee G)\\ \boxed{\begin{array}{c}\neg(\neg F \wedge \neg G)\\ \boxed{\begin{array}{c}\neg G\\ \neg\neg F\\ F\end{array}}\\ (F \vee G)\\ [\,]\end{array}}\\ (\neg F \wedge \neg G)\end{array}$$

- Lemmas shorten proofs, but enlarge the search space.

Natural Deduction - Soundness and Completeness

- Theorem 2.8: The calculus of natural deduction is sound and complete.
- Human readable proofs vs. machine generated proofs.
- Sequent calculus.
- Proof theory.

Sequent Calculus

- Sequent: $\mathcal{F} \vdash \mathcal{G}$, where $\mathcal{F}$ and $\mathcal{G}$ are multisets of formulas.
- Notation: $$\begin{array}{rcl} \dot{\{} F_1, \dots, F_n \dot{\}} & \rightsquigarrow & F_1, \dots, F_n. \\ \mathcal{F} \,\dot{\cup}\, \dot{\{} F \dot{\}} & \rightsquigarrow & \mathcal{F}, F. \end{array}$$
- Inference rules: $\dfrac{S_1 \dots S_n}{S}\,\mathsf{r}$ or $\dfrac{}{S}\,\mathsf{r}$

Sequent Calculus: Axiom, Cut and Structural Rules

Axiom

$$\frac{}{\boldsymbol{H}, \boldsymbol{\mathcal{F}} \vdash \boldsymbol{\mathcal{G}}, \boldsymbol{H}}\,\mathsf{ax}$$

Cut

$$\frac{\boldsymbol{\mathcal{F}} \vdash \boldsymbol{\mathcal{G}}, \boldsymbol{H} \quad \boldsymbol{H}, \boldsymbol{\mathcal{F}} \vdash \boldsymbol{\mathcal{G}}}{\boldsymbol{\mathcal{F}} \vdash \boldsymbol{\mathcal{G}}}\,\mathsf{cut}$$

Structural rules

$$\frac{\boldsymbol{H}, \boldsymbol{H}, \boldsymbol{\mathcal{F}} \vdash \boldsymbol{\mathcal{G}}}{\boldsymbol{H}, \boldsymbol{\mathcal{F}} \vdash \boldsymbol{\mathcal{G}}}\,\mathsf{cl} \qquad \frac{\boldsymbol{\mathcal{F}} \vdash \boldsymbol{\mathcal{G}}, \boldsymbol{H}, \boldsymbol{H}}{\boldsymbol{\mathcal{F}} \vdash \boldsymbol{\mathcal{G}}, \boldsymbol{H}}\,\mathsf{cr}$$

Sequent Calculus: Logical Rules

Rules for the left hand side	Rules for the right hand side
$$\dfrac{\mathcal{F} \vdash \mathcal{G}, H_1 \quad H_2, \mathcal{F} \vdash \mathcal{G}}{(H_1 \longrightarrow H_2), \mathcal{F} \vdash \mathcal{G}} \longrightarrow \mathsf{l}$$	$$\dfrac{H_1, \mathcal{F} \vdash \mathcal{G}, H_2}{\mathcal{F} \vdash \mathcal{G}, (H_1 \longrightarrow H_2)} \longrightarrow \mathsf{r}$$
$$\dfrac{H_1, H_2, \mathcal{F} \vdash \mathcal{G}}{(H_1 \wedge H_2), \mathcal{F} \vdash \mathcal{G}} \wedge\mathsf{l}$$	$$\dfrac{\mathcal{F} \vdash \mathcal{G}, H_1 \quad \mathcal{F} \vdash \mathcal{G}, H_2}{\mathcal{F} \vdash \mathcal{G}, (H_1 \wedge H_2)} \wedge\mathsf{r}$$
$$\dfrac{H_1, \mathcal{F} \vdash \mathcal{G} \quad H_2, \mathcal{F} \vdash \mathcal{G}}{(H_1 \vee H_2), \mathcal{F} \vdash \mathcal{G}} \vee\mathsf{l}$$	$$\dfrac{\mathcal{F} \vdash \mathcal{G}, H_1, H_2}{\mathcal{F} \vdash \mathcal{G}, (H_1 \vee H_2)} \vee\mathsf{r}$$
$$\dfrac{\mathcal{F} \vdash \mathcal{G}, H}{\neg H, \mathcal{F} \vdash \mathcal{G}} \neg\mathsf{l}$$	$$\dfrac{H, \mathcal{F} \vdash \mathcal{G}}{\mathcal{F} \vdash \mathcal{G}, \neg H} \neg\mathsf{r}$$

Sequent Calculus: Soundness and Completeness

- A proof of the sequent $\boldsymbol{S}$ in the sequent calculus is defined as follows:
  - An axiom of the form $\dfrac{}{\boldsymbol{S}}\,\mathsf{r}$ is a proof of $\boldsymbol{S}$.
  - If $\boldsymbol{T}_1, \ldots, \boldsymbol{T}_n$ are proofs of $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_n$ respectively and the sequent calculus contains a rule of the form $\dfrac{\boldsymbol{S}_1 \ldots \boldsymbol{S}_n}{\boldsymbol{S}}\,\mathsf{r}$, then $\dfrac{\boldsymbol{T}_1 \ldots \boldsymbol{T}_n}{\boldsymbol{S}}\,\mathsf{r}$ is a proof of $\boldsymbol{S}$.
- Theorem 2.9: The propositional sequent calculus is sound and complete.
- Subformula property: all formulas occurring in the condition of a rule occur as subformulas in the conclusion of the rule. $\rightsquigarrow$ Only the cut rule violates this property.
- Gentzen's "Hauptsatz": Cuts can be removed in sequent calculus proofs.

Resolution

- Let $C_1 = \{L, L_1, \ldots, L_n\}$ and $C_2 = \{\neg L, K_1, \ldots, K_m\}$ be two clauses. Then

$$\{K_1, \ldots, K_m, L_1, \ldots, L_n, \}$$

  is called a resolvent of C_1 and C_2.

  - $\{C_1, C_2\} \models \{K_1, \ldots, K_m, L_1, \ldots, L_n, \}$.

- A deduction of C from $\mathcal{F}$ is a finite sequence $C_1, \ldots, C_k = C$ of clauses such that C_i, $1 \leq i \leq k$, either is a clause from $\mathcal{F}$ or a resolvent of clauses preceding C_i.
- A deduction of [ ] from $\mathcal{F}$ is called a refutation.
- Theorem 2.11 The resolution calculus for propositional logic is sound and complete.

Monotonicity

- A logic is monotonic iff $\mathcal{F} \models F$ implies $\mathcal{F} \cup \mathcal{F}' \models F$.
    - Propositional logic is monotonic.

- A logic which obeys the law

$$\text{for each atom } p \text{ we find } \mathcal{F} \models p \text{ iff } p \in \mathcal{F}.$$

  is not monotonic.