Chapter 10
CLASSICAL AUTOMATED PROOF SYSTEMS

PART 3: GENTZEN SYSTEMS
Gentzen Sequent Calculus GL

The proof system GL for the classical propositional logic presented now is a version of the original Gentzen (1934) systems LK.

A constructive proof of the Completeness Theorem for the system GL is very similar to the proof of the Completeness Theorem for the system RS.

Expressions of the system are like in the original Gentzen system LK are Gentzen sequents.

Hence we use also a name Gentzen sequent calculus for it.
Gentzen Sequent Calculus GL

Language of GL

\[ \mathcal{L} = \mathcal{L}_{\{\cup, \cap, \Rightarrow, \neg\}} \]

We add a new symbol to the alphabet \(\rightarrow\) called a Gentzen arrow.

The sequents are built out of finite sequences (empty included) of formulas, i.e., elements of \(\mathcal{F}^*\), and the additional symbol \(\rightarrow\).

We denote, as in the RS system, the finite sequences of formulas by Greek capital letters

\[ \Gamma, \Delta, \Sigma, \ldots \]

with indices if necessary.
Gentzen Sequents

**Definition** Any expression

$$\Gamma \rightarrow \Delta$$

where $\Gamma, \Delta \in \mathcal{F}^*$ is called a sequent

Intuitively, we interpret semantically a sequent

$$A_1, ..., A_n \rightarrow B_1, ..., B_m$$

where $n, m \geq 1$, as a formula

$$(A_1 \cap ... \cap A_n) \Rightarrow (B_1 \cup ... \cup B_m)$$
Gentzen Sequents

The sequent

\[ A_1, ..., A_n \rightarrow \]

(where \( m \geq 1 \)) means that \( A_1 \cap ... \cap A_n \) yields a contradiction

The sequent

\[ \rightarrow B_1, ..., B_m \]

(where \( m \geq 1 \)) means semantically \( T \Rightarrow (B_1 \cup ... \cup B_m) \)

The empty sequent

\[ \rightarrow \]

means a contradiction
Gentzen Sequents

Given non empty sequences \( \Gamma, \Delta \)

We denote by \( \sigma_\Gamma \) any conjunction of all formulas of \( \Gamma \)

We denote by \( \delta_\Delta \) any disjunction of all formulas of \( \Delta \)

The intuitive semantics of a non-empty sequent \( \Gamma \rightarrow \Delta \) is

\[ \Gamma \rightarrow \Delta \equiv (\sigma_\Gamma \Rightarrow \delta_\Delta) \]
Formal Semantics

Formal semantics for sequents of $GL$ is defined as follows

Let $v : VAR \rightarrow \{T, F\}$ be a truth assignment and $v^*$ its extension to the set of formulas $\mathcal{F}$

We extend $v^*$ to the set

$$SQ = \{ \Gamma \rightarrow \Delta : \Gamma, \Delta \in \mathcal{F}^* \}$$

of all sequents as follows

For any sequent $\Gamma \rightarrow \Delta \in SQ$

$$v^*(\Gamma \rightarrow \Delta) = v^*(\sigma_{\Gamma}) \Rightarrow v^*(\delta_{\Delta})$$
Formal Semantics

In the case when $\Gamma = \emptyset$ or $\Delta = \emptyset$ we define

$$v^*(\Delta) = (T \Rightarrow v^*(\delta_{\Delta}))$$

$$v^*(\Gamma \rightarrow ) = (v^*(\sigma_{\Gamma}) \Rightarrow F)$$

The sequent $\Gamma \rightarrow \Delta$ is **satisfiable** if there is a truth assignment $v : VAR \rightarrow \{T, F\}$ such that

$$v^*(\Gamma \rightarrow \Delta) = T$$
Formal Semantics

**Model** for $\Gamma \rightarrow \Delta$ is any $v$ such that

$$v^* (\Gamma \rightarrow \Delta) = T$$

We write it $v \models \Gamma \rightarrow \Delta$

**Counter-model** is any $v$ such that

$$v^* (\Gamma \rightarrow \Delta) = F$$

We write it $v \not\models \Gamma \rightarrow \Delta$

**Tautology** is any sequent $\Gamma \rightarrow \Delta$ such that

$$v^* (\Gamma \rightarrow \Delta) = T \text{ for all truth assignments } v : VAR \rightarrow \{T, F\}$$

We write it $\models \Gamma \rightarrow \Delta$
Example

Let \( \Gamma \longrightarrow \Delta \) be a sequent

\[
a, (b \cap a) \longrightarrow \neg b, (b \Rightarrow a)
\]

The truth assignment \( v \) for which

\[
v(a) = T \quad \text{and} \quad v(b) = T
\]

is a model for \( \Gamma \longrightarrow \Delta \) as shows the following computation

\[
v^*(a, (b \cap a) \longrightarrow \neg b, (b \Rightarrow a)) = v^*(\sigma_{\{a,(b\cap a)\}}) \Rightarrow v^*(\delta_{\{\neg b,(b\Rightarrow a)\}})
\]

\[
= v(a) \cap (v(b) \cap v(a)) \Rightarrow \neg v(b) \cup (v(b) \Rightarrow v(a))
\]

\[
= T \cap T \cap T \Rightarrow \neg T \cup (T \Rightarrow T) = T \Rightarrow (F \cup T) = T \Rightarrow T = T
\]
Example

Observe that the truth assignment $v$ for which

$$v(a) = T \text{ and } v(b) = T$$

is the only one for which

$$v^* (\Gamma) = v^* (a, (b \cap a)) = T$$

and we proved that it is a model for

$$a, (b \cap a) \rightarrow \neg b, (b \Rightarrow a)$$

It is hence impossible to find $v$ which would falsify it, what proves that

$$\models a, (b \cap a) \rightarrow \neg b, (b \Rightarrow a)$$
Gentzen System **GL**

**Definition of GL**

**Logical Axioms LA**

We adopt as an axiom any sequent of variables (positive literals) which contains a propositional variable that appears on both sides of the sequent arrow $\rightarrow$, i.e. any sequent of the form

$$
\Gamma'_1, a, \Gamma'_2 \rightarrow \Delta'_1, a, \Delta'_2
$$

for any $a \in VAR$ and any sequences $\Gamma'_1, \Gamma'_2, \Delta'_1, \Delta'_2 \in VAR^*$.
Gentzen System \textbf{GL}

Inference rules of \textbf{GL}

Let \( \Gamma', \Delta' \in \text{VAR}^* \) and \( \Gamma, \Delta \in \mathcal{F}^* \)

Conjunction rules

\[
(\cap \rightarrow) \quad \frac{\Gamma', A, B, \Gamma \rightarrow \Delta'}{\Gamma', (A \cap B), \Gamma \rightarrow \Delta'}
\]

\[
(\rightarrow \cap) \quad \frac{\Gamma \rightarrow \Delta, A, \Delta' \quad \Gamma \rightarrow \Delta, B, \Delta'}{\Gamma \rightarrow \Delta, (A \cap B) \Delta'}
\]
Gentzen System **GL**

**Disjunction rules**

\[(\rightarrow \cup)\]

\[\frac{\Gamma \rightarrow \Delta, A, B, \Delta'}{\Gamma \rightarrow \Delta, (A \cup B), \Delta'}\]

\[(\cup \rightarrow)\]

\[\frac{\Gamma', A, \Gamma \rightarrow \Delta'; \Gamma', B, \Gamma \rightarrow \Delta'}{\Gamma', (A \cup B), \Gamma \rightarrow \Delta'}\]
Gentzen System \( GL \)

Implication rules

\[
\frac{\Gamma', A, \Gamma \rightarrow \Delta, B, \Delta'}{\Gamma', \Gamma \rightarrow \Delta, (A \Rightarrow B), \Delta'}
\]

\[
\frac{\Gamma', \Gamma \rightarrow \Delta, A, \Delta'; \Gamma', B, \Gamma \rightarrow \Delta, \Delta'}{\Gamma', (A \Rightarrow B), \Gamma \rightarrow \Delta, \Delta'}
\]
Gentzen System GL

Negation rules

\[ (\neg \rightarrow) \]
\[ \Gamma', \Gamma \rightarrow \Delta, A, \Delta' \]
\[ \Gamma', \neg A, \Gamma \rightarrow \Delta, \Delta' \]

\[ (\rightarrow \neg) \]
\[ \Gamma', A, \Gamma \rightarrow \Delta, \Delta' \]
\[ \Gamma', \Gamma \rightarrow \Delta, \neg A, \Delta' \]
We define the Gentzen System GL

\[
GL = ( L_{\{\cup, \cap, \Rightarrow, \neg\}}, \ SQ, \ LA, \ R )
\]

for

\[
R = \{(\cap \rightarrow), (\rightarrow \cap), (\cup \rightarrow), (\rightarrow \cup), (\Rightarrow \rightarrow), (\rightarrow \Rightarrow)\}
\]

\[
\cup \{(\neg \rightarrow), (\rightarrow \neg)\}
\]

We write, as usual,

\[
\vdash_{GL} \Gamma \rightarrow \Delta
\]

to denote that \( \Gamma \rightarrow \Delta \) has a formal proof in GL.

A formula \( A \in \mathcal{F} \), has a proof in GL if the sequent \( \rightarrow A \) has a proof in GL, i.e.

\[
\vdash_{GL} A \quad \text{if ad only if} \quad \rightarrow A
\]
Gentzen System \textbf{GL}

We consider, as we did with \textbf{RS} the proof trees for \textbf{GL}, i.e. we define

A \textbf{proof tree}, or \textbf{GL}-proof of $\Gamma \rightarrow \Delta$ is a tree

\[ T_{\Gamma \rightarrow \Delta} \]

of sequents satisfying the following conditions:

1. The topmost sequent, i.e. the root of $T_{\Gamma \rightarrow \Delta}$ is $\Gamma \rightarrow \Delta$

2. All \textbf{leafs} are axioms

3. The \textbf{nodes} are sequents such that each sequent on the tree follows from the ones immediately preceding it by one of the rules.
Gentzen System \text{GL}

Remark
The proof search in \text{GL} as defined by the decomposition tree for a given formula $A$ is \textbf{not always unique}

We show it on an example on the next slide
Example

A tree-proof in **GL** of the de Morgan Law

\[ \rightarrow (\neg(a \land b) \Rightarrow (\neg a \lor \neg b)) \]
\[
\mid (\rightarrow \Rightarrow)
\]
\[ \neg(a \land b) \rightarrow (\neg a \lor \neg b) \]
\[
\mid (\rightarrow \lor)
\]
\[ \neg(a \land b) \rightarrow \neg a, \neg b \]
\[
\mid (\rightarrow \neg)
\]
\[ b, \neg(a \land b) \rightarrow \neg a \]
\[
\mid (\rightarrow \neg)
\]
\[ b, a, \neg(a \land b) \rightarrow \]
\[
\mid (\neg \rightarrow \neg)
\]
\[ b, a \rightarrow (a \land b) \]
\[
\land (\rightarrow \land)
\]

\[ b, a \rightarrow a \]
\[ b, a \rightarrow b \]
Example

Here is another tree-proof in **GL** of the de Morgan Law

\[ \rightarrow (\neg(a \land b) \Rightarrow (\neg a \lor \neg b)) \]

\[ \left| (\rightarrow \Rightarrow) \right. \]
\[ \neg(a \land b) \rightarrow (\neg a \lor \neg b) \]

\[ \left| (\rightarrow \lor) \right. \]
\[ \neg(a \land b) \rightarrow \neg a, \neg b \]

\[ \left| (\rightarrow \neg) \right. \]
\[ b, \neg(a \land b) \rightarrow \neg a \]

\[ \left| (\neg \rightarrow) \right. \]
\[ b \rightarrow \neg a, (a \land b) \]

\[ \land (\rightarrow \land) \]

\[ b \rightarrow \neg a, a \]

\[ b \rightarrow \neg a, b \]

\[ \left| (\rightarrow \neg) \right. \]
\[ \neg a, a \]

\[ \left| (\rightarrow \neg) \right. \]
\[ \neg a, b \]

\[ b, a \rightarrow a \]

\[ b, a \rightarrow b \]
Gentzen System \textbf{GL}

The process of \textit{searching for proofs} of a formula \( A \) in \textbf{GL} consists, as in the \textbf{RS} type systems, of building certain trees, called \textit{decomposition trees}.

Their construction is similar to the one for \textbf{RS} type systems.

We take a \textbf{root} of a \textit{decomposition tree} \( T_A \) of of a formula \( A \) a sequent \( \rightarrow A \).

For each \textbf{node}, if there is a \textbf{rule} of \textbf{GL} which conclusion has the same form as \textbf{node} sequent, then the \textbf{node} has \textit{children} that are \textbf{premises} of the \textbf{rule}.

If the \textbf{node} consists only of a sequent built only out of \textbf{variables} then it \textit{does not} have any \textit{children}.

This is a \textit{termination condition} for the \textit{tree}.
Gentzen System GL

We prove that each formula $A$ generates a finite set of decomposition trees, $T_A$, such that the following holds:

If there exist a tree $T_A \in T_A$ whose all leafs are axioms, then tree $T_A$ constitutes a proof of $A$ in GL.

If all trees in $T_A$ have at least one non-axiom leaf, the proof of $A$ does not exist.

The first step in defining a notion of a decomposition tree consists of transforming the inference rules of GL, as we did in the case of the RS type systems, into corresponding decomposition rules.
Decomposition Rules of $\textbf{GL}$

Decomposition rules of $\textbf{GL}$
Let $\Gamma', \Delta' \in \text{VAR}^*$ and $\Gamma, \Delta \in \mathcal{F}^*$

Conjunction rules

\[
\begin{align*}
(\cap \rightarrow) & \quad \frac{\Gamma', (A \cap B), \Gamma \rightarrow \Delta'}{\Gamma', A, B, \Gamma \rightarrow \Delta'} \\
(\rightarrow \cap) & \quad \frac{\Gamma \rightarrow \Delta, (A \cap B) \Delta'}{\Gamma \rightarrow \Delta, A, \Delta'; \Gamma \rightarrow \Delta, B, \Delta'}
\end{align*}
\]
Decomposition Rules of GL

Disjunction rules

$$\Gamma \rightarrow \Delta, (A \cup B), \Delta'$$

$$\Gamma \rightarrow \Delta, A, B, \Delta'$$

$$\Gamma', (A \cup B), \Gamma \rightarrow \Delta'$$

$$\Gamma', A, \Gamma \rightarrow \Delta'; \Gamma', B, \Gamma \rightarrow \Delta'$$
Decomposition Rules of GL

Implication rules

\[(\rightarrow\rightleftharpoons) \quad \frac{\Gamma', \Gamma \rightarrow \Delta, (A \Rightarrow B), \Delta'}{\Gamma', A, \Gamma \rightarrow \Delta, B, \Delta'}\]

\[(\Rightarrow\rightleftharpoons) \quad \frac{\Gamma', (A \Rightarrow B), \Gamma \rightarrow \Delta, \Delta'}{\Gamma', \Gamma \rightarrow \Delta, A, \Delta'}; \quad \frac{\Gamma', B, \Gamma \rightarrow \Delta, \Delta'}{\Gamma', \Gamma \rightarrow \Delta, A, \Delta'}\]
Decomposition Rules of GL

Negation rules

\[
(\neg \rightarrow) \quad \frac{\Gamma', \neg A, \Gamma}{\Gamma', \Gamma} \rightarrow \Delta, \Delta'
\]

\[
(\rightarrow \neg) \quad \frac{\Gamma', \Gamma}{\Delta, \neg A, \Delta'}
\]

\[
\frac{\Gamma', A, \Gamma}{\Delta, \Delta'}
\]
Decomposition Tree Definition

For each formula $A \in \mathcal{F}$, a decomposition tree $T_A$ is a tree build as follows

**Step 1.** The sequent $\rightarrow A$ is the **root** of $T_A$

For any node $\Gamma \rightarrow \Delta$ of the tree we follow the steps below

**Step 2.** If $\Gamma \rightarrow \Delta$ is **indecomposable**, then $\Gamma \rightarrow \Delta$ becomes a **leaf** of the tree

**Step 3.** If $\Gamma \rightarrow \Delta$ is **decomposable**, then we pick a decomposition rule that **matches** the sequent of the **current** node

To do so we **proceed** as follows
1 Given a node $\Gamma \rightarrow \Delta$
We traverse $\Gamma$ from left to right to find the first decomposable formula
Its main connective $\circ$ identifies a possible decomposition rule ($\circ \rightarrow$) Then we check if this decomposition rule ($\circ \rightarrow$) applies. If it does we put its conclusions (conclusion) as leaves (leaf)
2 We traverse $\Delta$ from right to left to find the first decomposable formula
Its main connective $\circ$ identifies a possible decomposition rule ($\rightarrow \circ$) Then we check if this decomposition rule applies. If it does we put its conclusions (conclusion) as leaves (leaf)
3 If 1 and 2 applies we choose one of the rules
Step 4. We repeat Step 2. and Step 3. until we obtain only leaves
Decomposition Tree Definition

Observe that a decomposable $\Gamma \rightarrow \Delta$ is always in the domain in one of the decomposition rules $(\circ \rightarrow)$, $(\rightarrow \circ)$, or in the domain of both. Hence the tree $T_A$ may not be unique and all possible choices of 3. give all possible decomposition trees
Exercise
Prove, by constructing a proper decomposition tree that

$$
\vdash_{GL} (((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a)) \text{ if and only if }
\vdash_{GL} ((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))
$$

Solution
By definition, we have that

We construct a decomposition tree $T_{\rightarrow A}$ as follows
System GL Exercises

\[ T \rightarrow A \]

\[ \rightarrow (\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a) \]

| (\Rightarrow)

\[ (\neg a \Rightarrow b) \rightarrow (\neg b \Rightarrow a) \]

| (\Rightarrow)

\[ \neg b, (\neg a \Rightarrow b) \rightarrow a \]

| (\rightarrow \neg)

\[ (\neg a \Rightarrow b) \rightarrow b, a \]

\[ \bigwedge (\Rightarrow \rightarrow) \]

\[ \rightarrow \neg b, b, a \]

\[ b \rightarrow b, a \]

| (\rightarrow \neg)

\[ b \rightarrow b, a \]

axiom

\[ b \rightarrow b, a \]

axiom

All leaves of the tree are axioms, hence we have found the proof of A in GL
System \textbf{GL} Exercises

Exercise
Prove, by constructing proper \textit{decomposition trees} that

$$\not \vdash_{\text{GL}} ((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))$$

Solution
Observe that for any formula \( A \), its decomposition tree \( T_{\rightarrow A} \) in \textbf{GL} may \textbf{not be unique}

Hence we have to construct all possible \textit{decomposition trees} to prove that each of them has a \textbf{non axiom leaf}

We construct the decomposition trees for \( \rightarrow A \) as follows
The tree contains a non-axiom leaf, hence it is not a proof. We have one more tree to construct.
System **GL** Exercises

\[ T_1 \rightarrow A \]

\[ \rightarrow ((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a)) \]

\[ | (\rightarrow \Rightarrow) \ (one \ choice) \]

\[ (a \Rightarrow b) \rightarrow (\neg b \Rightarrow a) \]

\[ \wedge (\Rightarrow \rightarrow) \ (second \ choice) \]

\[ \rightarrow (\neg b \Rightarrow a), a \]

\[ | (\rightarrow \Rightarrow) \ (one \ choice) \]

\[ \neg b \rightarrow a, a \]

\[ | (\neg \rightarrow) \ (one \ choice) \]

\[ \rightarrow a, a, b \]

\[ non-axiom \]

\[ b \rightarrow (\neg b \Rightarrow a) \]

\[ | (\rightarrow \Rightarrow) \ (one \ choice) \]

\[ b, \neg b \rightarrow a \]

\[ | (\neg \rightarrow) \ (one \ choice) \]

\[ \neg b \rightarrow a, b \]

\[ axiom \]

All possible trees end with a non-axiom leaf. It proves that

\[ \kappa_G L ((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a)) \]
System GL Exercises

Does the tree below constitute a proof in GL? Justify your answer

\[
T \rightarrow A
\]

\[
\rightarrow \neg((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))
\]

\[
| (\rightarrow \neg)
\]

\[
\neg((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a)) \rightarrow
\]

\[
| (\neg \rightarrow)
\]

\[
(\neg a \Rightarrow b) \rightarrow (\neg b \Rightarrow a)
\]

\[
| (\rightarrow \Rightarrow)
\]

\[
(\neg a \Rightarrow b), \neg b \rightarrow a
\]

\[
| (\neg \neg)
\]

\[
(\neg a \Rightarrow b) \rightarrow b, a
\]

\[
\bigwedge (\Rightarrow \Rightarrow)
\]

\[
\rightarrow \neg a, b, a
\]

\[
b \rightarrow b, a
\]

\[
| (\rightarrow \neg)
\]

\[
a \rightarrow b, a
\]

\[
\text{axiom}
\]

\[
\text{axiom}
\]
Solution
The tree $T \rightarrow A$ is not a proof in GL because a rule corresponding to the decomposition step below does not exist in GL:

$$(\neg a \Rightarrow b), \neg b \rightarrow a$$

\[ | (\neg \rightarrow) \]

$$(\neg a \Rightarrow b) \rightarrow b, a$$

It is a proof is some system GL1 that has all the rules of GL except its rule $(\neg \rightarrow)$:

$$(\neg \rightarrow) \quad \Gamma', \Gamma \rightarrow \Delta, A, \Delta'$$

$$(\neg \rightarrow) \quad \Gamma', \neg A, \Gamma \rightarrow \Delta, \Delta'$$

This rule has to be replaced in by the rule:

$$(\neg \rightarrow)_1 \quad \Gamma, \Gamma' \rightarrow \Delta, A, \Delta'$$

$$(\Gamma, \neg A, \Gamma', \Gamma' \rightarrow \Delta, \Delta'$$
Exercises

Exercise 1
Write all proofs in $\text{GL}$ of

$((\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$

Exercise 2
Find a formula which has a unique decomposition tree

Exercise 3
Describe for which kind of formulas the decomposition tree is unique
We know that the system **GL** is strongly sound
Prove, by constructing a **counter-model** determined by a proper **decomposition tree** that

\[ \not\models ((b \Rightarrow a) \Rightarrow (\neg b \Rightarrow a)) \]

We construct the decomposition tree for the formula

**A** : \(((b \Rightarrow a) \Rightarrow (\neg b \Rightarrow a))\) as follows
System GL Exercises

\( T_{\rightarrow A} \)

\[ \rightarrow ((b \Rightarrow a) \Rightarrow (\neg b \Rightarrow a)) \]

\[ \mid (\rightarrow \Rightarrow) \]

\[ (b \Rightarrow a) \rightarrow (\neg b \Rightarrow a) \]

\[ \mid (\rightarrow \Rightarrow) \]

\[ \neg b, (b \Rightarrow a) \rightarrow a \]

\[ \mid (\neg \rightarrow) \]

\[ (b \Rightarrow a) \rightarrow b, a \]

\[ \bigwedge (\Rightarrow \rightarrow) \]

\[ \rightarrow b, b, a \]

\[ a \rightarrow b, a \]

non – axiom \hfill axiom

The **counter model** determined by the tree \( T_{\rightarrow A} \) is any truth assignment \( v \) that evaluates the **non axiom leaf** \( \rightarrow b, b, a \) to \( F \)
Let $v : \text{VAR} \rightarrow \{T, F\}$ be a truth assignment.

By definition of semantic for sequents we have that $v^*(\rightarrow b, b, a) = (T \Rightarrow v(b) \cup v(b) \cup v(a))$.

Hence $v^*(\rightarrow b, b, a) = F$ if and only if $(T \Rightarrow v(b) \cup v(b) \cup v(a)) = F$ if and only if $v(b) = v(a) = F$.

The **counter model** determined by the $T \rightarrow A$ is any $v : \text{VAR} \rightarrow \{T, F\}$ such that $v(b) = v(a) = F$. 
Exercise
Prove the **Completeness theorem** for GL
Assume that the **Strong Soundness** has been already proved and the **Decompositions Trees** are already defined

Reminder

**Formula Completeness** for GL: For any \( A \in \mathcal{F} \),

\[
\models A \quad \text{if and only if} \quad \vdash_{GL} \rightarrow A
\]

**Soundness** for GL: For any \( A \in \mathcal{F} \),

If \( \vdash_{GL} \rightarrow A \), then \( \models A \)

**Completeness part** for GL: For any \( A \in \mathcal{F} \),

If \( \models A \), then \( \vdash_{GL} \rightarrow A \)
Proof of Completeness of \textit{GL}

We prove the logically equivalent form of the \textbf{Completeness part}: For any $A \in F$, 

$$\text{If } \kappa_{GL} \rightarrow A \text{ then } \not\models A$$

\textbf{Proof}

Assume $\kappa_{GL} \rightarrow A$, i.e. $\rightarrow A$ does not have a proof in \textit{GL}

Let $\mathcal{T}_A$ be a \textbf{set of all decomposition trees} of $\rightarrow A$

As $\kappa_{GL} \rightarrow A$, each $T \in \mathcal{T}_A$ has a \textbf{non-axiom leaf}

We choose an arbitrary $T_A \in \mathcal{T}_A$
Proof of Completeness of GL

Let $\Gamma' \rightarrow \Delta'$, $\Gamma'$, $\Delta' \in \text{VAR}^*$ be the non-axiom leaf of the tree $T_A$

The non-axiom leaf $\Gamma' \rightarrow \Delta'$ determines a truth assignment $v : \text{VAR} \rightarrow \{T, F\}$ which is defined as follows:

$$v(a) = \begin{cases} T & \text{if } a \text{ appears in } \Gamma' \\ F & \text{if } a \text{ appears in } \Delta' \\ \text{any value} & \text{if } a \text{ does not appear in } \Gamma' \rightarrow \Delta' \\ \end{cases}$$

By the strong soundness of the rules of inference of GL it proves that $v^*(A) = F$, i.e. that $\not\models A$
Original Gentzen systems

**LK** for Classical Propositional Logic and **LI** for Intuitionistic Logic
Gentzen Systems  **LK, LI**  for Classical and Intuitionistic Logics

The proof systems  **LK**  for Classical Propositional Logic and  **LI**  for Intuitionistic Propositional Logic as presented here were published by  G. Gentzen  in 1935

The proof system  **LI**  for Intuitionistic Logic was presented as a particular case of his proof system  **LK**  for the classical logic.
Classical Gentzen System $\textbf{LK}$

**Language of LK**

$$\mathcal{L} = \mathcal{L}\{\neg, \land, \lor, \Rightarrow\} \quad \text{and} \quad \mathcal{E} = SQ$$

for

$$SQ = \{\Gamma \rightarrow \Delta : \Gamma, \Delta \in \mathcal{F}^*\}$$

**Axioms of LK** any sequent of the form

$$\Gamma_1, A, \Gamma_2 \rightarrow \Gamma_3, A, \Gamma_4$$
Classical Gentzen System **LK**

Rules of inference of **LK** are as follows

**Structural Rules**

**Weakening**

(\textit{weak} \rightarrow)

\[
\frac{\Gamma \rightarrow \Delta}{A, \Gamma \rightarrow \Delta}
\]

(\rightarrow \textit{weak})

\[
\frac{\Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta, A}
\]

**Contraction**

(\textit{contr} \rightarrow)

\[
\frac{A, A, \Gamma \rightarrow \Delta}{A, \Gamma \rightarrow \Delta}
\]

(\rightarrow \textit{contr})

\[
\frac{\Gamma \rightarrow \Delta, A, A}{\Gamma \rightarrow \Delta, A}
\]
Classical Gentzen System \textbf{LK}

\section*{Structural Rules}

\subsection*{Exchange}

\begin{align*}
\text{\texttt{(exch \rightarrow)}} & \quad \frac{\Gamma_1, A, B, \Gamma_2 \to \Delta}{\Gamma_1, B, A, \Gamma_2 \to \Delta} \\
\text{\texttt{(\rightarrow exch)}} & \quad \frac{\Delta \to \Gamma_1, A, B, \Gamma_2}{\Delta \to \Gamma_1, B, A, \Gamma_2}
\end{align*}
Classical Gentzen System LK

Logical Rules

Conjunction rules

\[
(\cap \rightarrow) \quad \frac{A, B, \Gamma \rightarrow \Delta}{(A \cap B), \Gamma \rightarrow \Delta}
\]

\[
(\rightarrow \cap) \quad \frac{\Gamma \rightarrow \Delta, A}{\Gamma \rightarrow \Delta, (A \cap B)}
\]

Disjunction rules

\[
(\rightarrow \cup) \quad \frac{\Gamma \rightarrow \Delta, A, B}{\Gamma \rightarrow \Delta, (A \cup B)}
\]

\[
(\cup \rightarrow) \quad \frac{A, \Gamma \rightarrow \Delta ; B, \Gamma \rightarrow \Delta}{(A \cup B), \Gamma \rightarrow \Delta}
\]
Classical Gentzen System \textbf{LK}

Implication rules

\begin{align*}
\quad & (\rightarrow\rightarrow) \quad \frac{A, \Gamma \rightarrow \Delta, B}{\Gamma \rightarrow \Delta, (A \Rightarrow B)} \\
\quad & (\Rightarrow\rightarrow) \quad \frac{\Gamma \rightarrow \Delta, A ; B, \Gamma \rightarrow \Delta}{(A \Rightarrow B), \Gamma \rightarrow \Delta}
\end{align*}
Classical Gentzen System \( \textbf{LK} \)

**Negation rules**

\[
\begin{align*}
(\neg \rightarrow) & \quad \frac{\Gamma \rightarrow \Delta, A}{\neg A, \Gamma \rightarrow \Delta} \\
(\rightarrow \neg) & \quad \frac{A, \Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta, \neg A}
\end{align*}
\]

**We define** formally

\( \textbf{LK} = (\mathcal{L}, \text{SQ}, \text{AX}, \text{ Structural rules, Logical rules}) \)
Intuitionistic I Gentzen System $\mathbf{LI}$

Language of $\mathbf{LI}$

Any expression

$$\Gamma \rightarrow \Delta$$

where $\Gamma, \Delta \in \mathcal{F}^*$ and

$$\Delta \quad \text{consists of at most one formula}$$

is called a $\mathbf{LI}$ sequent

We denote the set of all $\mathbf{LI}$ sequents by $\mathbf{ISQ}$, i.e.

$$\mathbf{ISQ} = \{ \Gamma \rightarrow \Delta : \Delta \quad \text{consists of at most one formula} \}$$
Axioms of LI

Axioms of LI consist of any sequent from the set $ISQ$ which contains a formula that appears on both sides of the sequent arrow $\rightarrow$, i.e. any sequent of the form

$$\Gamma, A, \Delta \rightarrow A$$

for $\Gamma, \Delta \in \mathcal{F}^*$
Rules of Inference of LI

The set inference rules of LI is divided into two groups: the structural rules and the logical rules. There are three Structural Rules of LI: Weakening, Contraction, and Exchange.

Weakening structural rule

\[
\begin{align*}
\text{(weak } \to) & \quad \frac{\Gamma \to \Delta}{A, \Gamma \to \Delta} \\
\text{(} \to \text{ weak)} & \quad \frac{}{\Gamma \to A}
\end{align*}
\]

A is called the weakening formula.

Remember that $\Delta$ contains at most one formula.
Rules of Inference of LI

Contraction structural rule

\[(\text{contr} \rightarrow) \quad \frac{A, A, \Gamma \rightarrow \Delta}{A, \Gamma \rightarrow \Delta}\]

The case below is **not VALID** for LI; we list it as it will be used in the classical case

\[(\rightarrow \text{contr}) \quad \frac{\Gamma \rightarrow \Delta, A, A}{\Gamma \rightarrow \Delta, A}\]

\(A\) is called the contraction formula

**Remember** that \(\Delta\) contains at most one formula
Rules of Inference of \( \text{LI} \)

**Exchange structural rule**

\[
\text{(exch } \rightarrow \text{)} \quad \frac{\Gamma_1, A, B, \Gamma_2 \rightarrow \Delta}{\Gamma_1, B, A, \Gamma_2 \rightarrow \Delta}
\]

The case below is **not VALID** for \( \text{LI} \); we list it as it will be used in the classical case

\[
\text{(} \rightarrow \text{ exch)} \quad \frac{\Delta \rightarrow \Gamma_1, A, B, \Gamma_2}{\Delta \rightarrow \Gamma_1, B, A, \Gamma_2}.
\]

**Remember** that \( \Delta \) contains at most one formula.
Rules of Inference of LI

Logical Rules

Conjunction rules

\[
\begin{align*}
(\cap \to) \quad & A, B, \Gamma \to \Delta \\
& (A \cap B), \Gamma \to \Delta', \\
(\to \cap) \quad & \Gamma \to A ; \Gamma \to B \\
& \Gamma \to (A \cap B)
\end{align*}
\]

Remember that $\Delta$ contains at most one formula
Rules of Inference of LI

Disjunction rules

1. \((\rightarrow \cup)_1\)
   \[ \Gamma \rightarrow A \quad \Rightarrow \quad \Gamma \rightarrow (A \cup B) \]

2. \((\rightarrow \cup)_2\)
   \[ \Gamma \rightarrow B \quad \Rightarrow \quad \Gamma \rightarrow (A \cup B) \]

3. \((\cup \rightarrow)\)
   \[ A, \Gamma \rightarrow \Delta \quad ; \quad B, \Gamma \rightarrow \Delta \quad \Rightarrow \quad (A \cup B), \Gamma \rightarrow \Delta \]

Remember that \(\Delta\) contains at most one formula