Chapter 11: Automated Proof Systems

SYSTEM RS OVERVIEW

Hilbert style systems are easy to define and admit a simple proof of the Completeness Theorem but they are difficult to use.

Automated systems are less intuitive then the Hilbert-style systems, but they will allow us to give an effective automatic procedure for proof search, what was impossible in a case of the Hilbert-style systems.

The first idea of this type was presented by **G. Gentzen** in 1934.

PART 1: RS SYSTEM

RS proof system presented here is due to **H. Rasiowa and R. Sikorski** and appeared for the first time in 1961. It extends easily to Predicate Logic and admits a CONSTRUCTIVE proof of Completeness Theorem (first given by Rasiowa- Sikorski).

PART 2: GENTZEN SYSTEM

We present two Gentzen Systems; a modern version and the original version. BOTH extend easily to Predicate Logic and admit a CONSTRUCTIVE proof of Completeness Theorem via Rasiowa-Sikorski method. The Original Gentzen system is easily adopted to a complete system foir the Intuitionistic Logic and will be presented in Chapter 12.

Language of **RS** is

$$\mathcal{L} = \mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}}.$$

The rules of inference of our system **RS** operate on *finite sequences of formulas*.

Set of expressions $\mathcal{E} = \mathcal{F}^*$.

Notation: elements of $\mathcal{E}$ are finite sequences of formulas and we denote them by

$$\Gamma, \Delta, \Sigma,$$

with indices if necessary.

Meaning of Sequences: the intuitive meaning of a sequence $\Gamma \in \mathcal{F}^*$ is that the truth assignment v makes it true if and only if it makes the formula of the form of the disjunction of all formulas of Γ true.

For any sequence $\Gamma \in \mathcal{F}^*$,

$$\Gamma = A_1, A_2, ..., A_n$$

we define

$$\delta_\Gamma = A_1 \cup A_2 \cup ... \cup A_n.$$

Formal Semantics for RS Let $v : VAR \longrightarrow \{T, F\}$ be a truth assignment, v^* its classical semantics extension to the set of formulas $\mathcal{F}$.

We formally extend v to the set $\mathcal{F}^*$ of all finite sequences of $\mathcal{F}$ as follows.

$$v^*(\Gamma) = v^*(\delta_\Gamma) = v^*(A_1) \cup v^*(A_2) \cup ... \cup v^*(A_n).$$

Model The sequence Γ is said to be *satisfiable* if there is a truth assignment $v : VAR \longrightarrow \{T, F\}$ such that $v^*(\Gamma) = T$.

Such a truth assignment v is called *a model* for Γ.

Counter- Model The sequence Γ is said to be *falsifiable* if there is a truth assignment v, such that $v^*(\Gamma) = F$.

Such a truth assignment v is also called *a counter-model* for Γ.

Tautology The sequence Γ is said to be *a tautology* if $v^*(\Gamma) = T$ for all truth assignments $v : VAR \longrightarrow \{T, F\}$.

Example Let Γ be a sequence

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

The truth assignment v for which $v(a) = F$ and $v(b) = T$ falsifies Γ, i.e. is *a counter-model* for Γ, as shows the following computation.

$v^*(\Gamma) = v^*(\delta_\Gamma) = v^*(a) \cup v^*(b \cap a) \cup v^*(\neg b) \cup v^*(b \Rightarrow a) = F \cup (F \cap T) \cup F \cup (T \Rightarrow F) = F \cup F \cup F \cup F = F$.

Rules of inference of **RS** are of the form:

$$\frac{\Gamma_1}{\Gamma} \quad or \quad \frac{\Gamma_1 \ ; \ \Gamma_2}{\Gamma},$$

where Γ_1, Γ_2 and Γ are sequences Γ_1, Γ_2 are called premisses and Γ is called the conclusion of the rule of inference.

Each rule of inference introduces a new logical connective, or a negation of a logical connective.

We name the rule that introduces the logical connective $\circ$ in the conclusion sequent Γ by $(\circ)$.

The notation $(\neg\circ)$ means that the negation of the logical connective $\circ$ is introduced in the conclusion sequence Γ.

System RS contains seven inference rules:

$$(\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg \Rightarrow), \text{ and } (\neg\neg).$$

Before we define the rules of inference of **RS** we need to introduce some definitions.

Literals

$$LT = VAR \cup \{\neg a : a \in VAR\}.$$

The variables are called *positive literals*.

Negations of variables are called *negative literals*.

We denote by $\Gamma^{'}$, $\Delta^{'}$, $\Sigma^{'}$ finite sequences (empty included) formed out of *literals* i.e

$$\Gamma^{'}, \Delta^{'}, \Sigma^{'} \in LT^{*}.$$

We will denote by Γ, Δ, Σ the elements of $\mathcal{F}^{*}$.

Axioms $\mathcal{AL}$ of RS We adopt as an axiom any sequence which contains any propositional variable and its negation, i.e any sequence

$$\Gamma_1^{'}, a, \Gamma_2^{'}, \neg a, \Gamma_3^{'},$$

$$\Gamma_1^{'}, \neg a, \Gamma_2^{'}, a, \Gamma_3^{'}.$$

Inference rules of RS

Disjunction rules

$$(\cup)\ \frac{\Gamma', A, B, \Delta}{\Gamma', (A \cup B), \Delta}, \qquad (\neg\cup)\ \frac{\Gamma', \neg A, \Delta \ : \ \Gamma', \neg B, \Delta}{\Gamma', \neg(A \cup B), \Delta}$$

Conjunction rules

$$(\cap)\ \frac{\Gamma', A, \Delta \ ; \ \Gamma', B, \Delta}{\Gamma', (A \cap B), \Delta}, \qquad (\neg\cap)\ \frac{\Gamma', \neg A, \neg B, \Delta}{\Gamma', \neg(A \cap B), \Delta}$$

Implication rules

$$(\Rightarrow)\ \frac{\Gamma', \neg A, B, \Delta}{\Gamma', (A \Rightarrow B), \Delta}, \quad (\neg \Rightarrow)\ \frac{\Gamma', A, \Delta : \Gamma', \neg B, \Delta}{\Gamma', \neg(A \Rightarrow B), \Delta}$$

Negation rule

$$(\neg\neg)\ \frac{\Gamma', A, \Delta}{\Gamma', \neg\neg A, \Delta}$$

where $\Gamma' \in LT^*, \Delta \in \mathcal{F}^*, A, B \in \mathcal{F}$.

The Proof System RS Formally we define:

$$\mathbf{RS} = (\mathcal{L}, \mathcal{E}, \mathcal{AL}, (\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg \Rightarrow), (\neg\neg))$$

Proof Tree By a proof tree, or **RS**-proof of Γ we understand a tree $\mathbf{T}_\Gamma$ of sequences satisfying the following conditions:

1. The topmost sequence, i.e *the root* of $\mathbf{T}_\Gamma$ is Γ,

2. all *leafs* are axioms,

3. the *nodes* are sequences such that each sequence on the tree follows from the ones immediately preceding it by one of the rules.

We picture, and write our proof trees with the node on the top, and leafs on the very bottom, instead of more common way, where the leafs are on the top and root is on the bottom of the tree.

We write our proof trees indicating additionally the name of the inference rule used at each step of the proof.

For example, if the proof of a *theorem* from *three axioms* was obtained by the subsequent use of the rules $(\cap), (\cup), (\cup), (\cap), (\cup)$, and $(\neg\neg)$, $(\Rightarrow)$,

We represent it as the following tree:

$$
\begin{array}{c}
theorem; \quad provable \quad formula\\
\mid (\Rightarrow)\\
conclusion\ of\ (\neg\neg)\\
\mid (\neg\neg)\\
conclusion\ of\ (\cup)\\
\mid (\cup)\\
conclusion\ of\ (\cap)\\
\bigwedge(\cap)\\
\begin{array}{cc}
conclusion\ of\ (\cap) & conclusion\ of\ (\cup)\\
\mid (\cup) & \mid (\cup)\\
axiom & conclusion\ of\ (\cap)\\
 & \bigwedge(\cap)\\
 & axiom \quad axiom
\end{array}
\end{array}
$$

Trees represent a certain *visualization* for the proofs and any formal proof in any system can be represented in a tree form.

Example The proof tree in **RS** of the de Morgan law

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

is the following.

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

$$\mid (\Rightarrow)$$

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

$$\mid (\neg\neg)$$

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

$$\bigwedge(\cap)$$

$$a, (\neg a \cup \neg b) \quad b, (\neg a \cup \neg b)$$

$$\mid (\cup) \qquad\qquad \mid (\cup)$$

$$a, \neg a, \neg b \qquad b, \neg a, \neg b$$

To obtain a "linear " formal proof (written in a vertical form) of it we just write down the tree as a sequence, starting from the leafs and going up (from left to right) to the root.

$$a, \neg a, \neg b$$

$$b, \neg a, \neg b$$

$$a, (\neg a \cup \neg b)$$

$$b, (\neg a \cup \neg b$$

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

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

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

The search for the proof of $(\neg(a \cup b) \Rightarrow (\neg a \cap \neg b))$ consists of building a certain tree and proceeds as follows.

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

$$\mid (\Rightarrow)$$

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

$$\mid (\neg\neg)$$

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

$$\mid (\cup)$$

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

$$\bigwedge(\cap)$$

$$a, b, \neg a \qquad\qquad a, b, \neg b$$

We construct its formal proof, written in a vertical manner, by writing the two axioms, which form the two premisses of the rule $(\cap)$ one above the other. All other sequences remain the same.

$$a, b, \neg b$$

$$a, b, \neg a$$

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

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

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

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

The tree generated by the proof search is called *a decomposition tree*.

Example of decomposition tree for $(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$ is the following.

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

$$(a \Rightarrow b), (a \Rightarrow c) \qquad \neg c, (a \Rightarrow c)$$

$$\mid (\Rightarrow) \qquad\qquad \mid (\Rightarrow)$$

$$\neg a, b, (a \Rightarrow c) \qquad \neg c, \neg a, c$$

$$\mid (\Rightarrow)$$

$$\neg a, b, \neg a, c$$

The decomposition tree generated by this search contains an non-axiom leaf and hence is not a proof.

Moreover, it proves, as the decomposition (proof search) tree is unique that the proof of the formula

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

DOES not EXIST in the system **RS**.

Counter-model generated by the decomposition tree.

Example: Given a formula A:

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

and its decomposition tree $\mathbf{T}_A$.

$$
\begin{array}{c}
(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c)) \\
\mid (\cup) \\
((a \Rightarrow b) \cap \neg c), (a \Rightarrow c) \\
\bigwedge (\cap) \\
\begin{array}{cc}
(a \Rightarrow b), (a \Rightarrow c) & \neg c, (a \Rightarrow c) \\
\mid (\Rightarrow) & \mid (\Rightarrow) \\
\neg a, b, (a \Rightarrow c) & \neg c, \neg a, c \\
\mid (\Rightarrow) & \\
\neg a, b, \neg a, c &
\end{array}
\end{array}
$$

Consider a non-axiom leaf:

$$\neg a, b, \neg a, c$$

Let v be any variable assignment

$$v: \; VAR \longrightarrow \{T, F\}$$

such that it makes this non-axiom leaf False, i.e. we put

$$v(a) = T, v(b) = F, v(c) = F.$$

Obviously, we have that

$$v^*(\neg a, b, \neg a, c) = F.$$

Moreover, all our rules of inference are sound (to be proven formally in the next section).

Rules soundness means that if one of premisses of a rule is FALSE, so is the conclusion.

Hence, the soundness of the rules proves (by induction on the degree of sequences $\Gamma \in \mathbf{T}_A$) that v, as defined above falsifies all sequences on the branch of $\mathbf{T}_A$ that ends with the non-axiom leaf $\neg a, b, \neg a, c$.

In particular, the formula A is on this branch, hence

$$v^*(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c)) = F$$

and v is *a counter-model* for A.

The truth assignments defined by a non- axiom leaves are called **counter-models generated** by the decomposition tree.

The construction of the counter-models generated by the decomposition trees are crucial to the proof of the Completeness Theorem for **RS**.

We prove first the following Completeness Theorem for formulas $A \in \mathcal{F}$,

Completeness Theorem 1 For any formula $A \in \mathcal{F}$,

$\vdash_{\mathbf{RS}} A$ if and only if $\models A$.

and then we deduce from it the following full Completeness Theorem for sequences $\Gamma \in \mathcal{F}^*$.

Completeness Theorem 2

For any $\Gamma \in \mathcal{F}^*$,

$\vdash_{\mathbf{RS}} \Gamma$ if and only if $\models \Gamma$.

The Completeness Theorem consists of two parts:

Soundness Part: (Soundness Theorem) for any $A \in \mathcal{F}$,

if $\vdash_{\mathbf{RS}} A$, then $\models A$.

Completeness Part: For any formula $A \in \mathcal{F}$,

if $\models A$, then $\vdash_{\mathbf{RS}} A$.

Soundness Theorem for RS

For any $\Gamma \in \mathcal{F}^*$,

if $\vdash_{\mathbf{RS}} \Gamma$, then $\models \Gamma$.

In particular, for any $A \in \mathcal{F}$,

if $\vdash_{\mathbf{RS}} A$, then $\models A$.

We prove as an example the soundness of two of inference rules: $(\Rightarrow)$ and $(\neg\cup)$ of **G**.

We show even more, that the premisses and conclusion of both rules are logically equivalent.

If $P_1, (P_2)$ are premiss(es) of a rule, C is its conclusion, then

$$v^*(P_1) = v^*(C)$$

in case of one premiss rule and

$$v^*(P_1) \cap v^*(P_2) = v^*(C),$$

in case of the two premisses rule.

Consider the rule $(\cup)$.

$$(\cup) \quad \frac{\Gamma', A, B, \Delta}{\Gamma', (A \cup B), \Delta}.$$

We evaluate: $v^*(\Gamma', A, B, \Delta) = v^*(\delta_{\{\Gamma', A, B, \Delta\}}) = v^*(\Gamma') \cup v^*(A) \cup v^*(B) \cup v^*(\Delta) = v^*(\Gamma') \cup v^*(A \cup B) \cup v^*(\Delta) = v^*(\delta_{\{\Gamma', (A\cup B), \Delta\}}) = v^*(\Gamma', (A \cup B), \Delta)$.

Consider the rule $(\neg\cup)$.

$$(\neg\cup) \quad \frac{\Gamma^{'}, \neg A, \Delta \ : \ \Gamma^{'}, \neg B, \Delta}{\Gamma^{'}, \neg(A \cup B), \Delta}.$$

We evaluate: $v^*(\Gamma^{'}, \neg A, \Delta) \cap v^*(\Gamma^{'}, \neg B, \Delta) = (v^*(\Gamma^{'}) \cup v^*(\neg A) \cup v^*(\Delta)) \cap (v^*(\Gamma^{'}) \cup v^*(\neg B) \cup v^*(\Delta)) = (v^*(\Gamma^{'}, \Delta) \cup v^*(\neg A)) \cap (v^*(\Gamma^{'}, \Delta) \cup v^*(\neg B)) =$ by distributivity $= (v^*(\Gamma^{'}, \Delta) \cup (v^*(\neg A) \cap v^*(\neg B)) = v^*(\Gamma^{'}) \cup v^*(\Delta) \cup (v^*(\neg A \cap \neg B)) =$ by the logical equivalence of $(\neg A \cap \neg B)$ and $\neg(A \cup B)$ $= v^*(\delta_{\{\Gamma', \neg(A \cup B), \Delta\}} = v^*(\Gamma^{'}, \neg(A \cup B), \Delta))$.

We prove now the Completeness Part of the Completeness Theorem:

If $\nvdash_{\mathbf{RS}} A$, then $\not\models A$.

STEPS needed for proof:

Step 1 Define, for each $A \in \mathcal{F}$ its *decomposition tree* $\mathbf{T}_A$.

Step 2 (Lemma 1) Prove that the decomposition tree $\mathbf{T}_A$ is *unique*.

Step 3 (Lemma 2) Prove that $\mathbf{T}_A$ has the following property:

Proof of A in **RS** does not exist ($\nvdash_{\mathbf{RS}}$)A, iff there is a leaf of $\mathbf{T}_A$ which is not an axiom.

Step 4 (Lemma 3) Prove that given A with $\mathbf{T}_A$ with a non-axiom leaf, we have that for any truth assignment v, such that v^*(non-axiom leaf) $= F$, v also falsifies A, i.e.

$$v^*(A) = F.$$

Proof of Completeness: Assume that A is any formula is such that

$$\nvdash_{\mathbf{RS}} A.$$

By the STEP 3, the decomposition tree $\mathbf{T}_A$ contains a non-axiom leaf.

The non-axiom leaf $\mathcal{L}_A$ *defines* a truth assignment v which falsifies A, as follows:

$$v(a) = \begin{cases} F & \text{if a appears in } \mathcal{L}_A \\ T & \text{if } \neg a \text{ appears in } \mathcal{L}_A \\ any\ value & \text{if a does not appear in } \mathcal{L}_A \end{cases}$$

This proves by STEP 4 that $\not\models A$.

RS: DECOMPOSITION TREES

The process of searching for the proof of a formula A in **RS** consists of building a certain tree, called a **decomposition tree** whose root is the formula A, nodes correspond to sequences which are conclusions of certain rules (and those rules are well defined at each step by the way the node is built), and leafs are axioms or are sequences of a non- axiom literals.

We prove that each formula A (sequence Γ) generates its *unique and finite* decomposition tree, $\mathbf{T}_A$ ($\mathbf{T}_\Gamma$).

The tree constitutes the proof of A (Γ) in **RS** if all its leafs are axioms.

If there is a leaf of $\mathbf{T}_A$ ($\mathbf{T}_\Gamma$) that *is not an axiom*, the tree is not a proof, moreover, the proof of A **does not exist**.

Before we give a proper definition of the proof search procedure by building a decomposition tree we list few important observations about the structure of the rules of the system **RS**.

Introduction of Connectives

The rules of **RS** are defined in such a way that each of them *introduces* a new logical connective, or a negation of a connective to a sequence in its domain (rules $(\cup), (\Rightarrow), (\cap)$) or a negation of a new logical connective (rules $(\neg\cup), (\neg\cap), (\neg \Rightarrow), (\neg\neg)$).

The rule $(\cup)$ introduces a new connective $\cup$ to a sequence Γ', A, B, Δ and it becomes, after the application of the rule, a sequence $\Gamma', (A \cup B), \Delta$.

Hence a name for this rule is $(\cup)$.

The rule $(\neg\cup)$ introduces a negation of a connective, $\neg\cup$ by combining sequences $\Gamma^{'}, \neg A, \Delta$ and $\Gamma^{'}, \neg B, \Delta$ into one sequence (conclusion of the rule) $\Gamma^{'}, \neg(A \cup B), \Delta$.

Hence a name for this rule is $(\neg\cup)$.

The same applies to all remaining rules of **RS**, hence their names say which connective, or the negation of which connective has been introduced by the particular rule.

Decomposition Rules

Building decomposition tree (a proof search tree) consists of using the inference rules in an inverse order; we transform them into rules that transform a conclusion into its premisses.

We call such rules the **decomposition rules**.

Here are all of **RS** decomposition rules.

Disjunction decomposition rules

$$(\cup)\ \frac{\Gamma^{'}, (A \cup B), \Delta}{\Gamma', A, B, \Delta}, \qquad (\neg\cup)\ \frac{\Gamma^{'}, \neg(A \cup B), \Delta}{\Gamma', \neg A, \Delta \ \ : \ \ \Gamma', \neg B, \Delta}$$

Conjunction decomposition rules

$$(\cap)\ \frac{\Gamma^{'}, (A \cap B), \Delta}{\Gamma', A, \Delta \ ; \ \Gamma', B, \Delta}, \qquad (\neg\cap)\ \frac{\Gamma^{'}, \neg(A \cap B), \Delta}{\Gamma', \neg A, \neg B, \Delta}$$

Implication decomposition rules

$$(\Rightarrow)\ \frac{\Gamma^{'}, (A \Rightarrow B), \Delta}{\Gamma', \neg A, B, \Delta}, \qquad (\neg \Rightarrow)\ \frac{\Gamma^{'}, \neg(A \Rightarrow B), \Delta}{\Gamma', A, \Delta \ \ : \ \ \Gamma', \neg B, \Delta}$$

Negation decomposition rule

$$(\neg\neg)\ \frac{\Gamma^{'}, \neg\neg A, \Delta}{\Gamma', A, \Delta}$$

where $\Gamma^{'} \in \mathcal{F}'^{*}, \Delta \in \mathcal{F}^{*}, A, B \in \mathcal{F}$.

We write the decomposition rules in **a visual tree form** as follows.

Tree Decomposition Rules

($\cup$) **rule:**

$$\Gamma^{'}, (A \cup B), \Delta$$

$$\mid (\cup)$$

$$\Gamma^{'}, A, B, \Delta$$

$(\neg\cup)$ **rule:**

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

$$\bigwedge (\neg\cup)$$

$$\Gamma', \neg A, \Delta \quad \Gamma', \neg B, \Delta$$

$(\cap)$ **rule:**

$$\Gamma', (A \cap B), \Delta$$

$$\bigwedge (\cap)$$

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

$(\neg\cup)$ **rule:**

$$\Gamma', \neg(A \cap B), \Delta$$

$$\mid (\neg\cap)$$

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

$(\Rightarrow)$ **rule:**

$$\Gamma', (A \Rightarrow B), \Delta$$

$$\mid (\cup)$$

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

$(\neg \Rightarrow)$ **rule:**

$$\Gamma', \neg(A \Rightarrow B), \Delta$$

$$\bigwedge(\neg \Rightarrow)$$

$$\Gamma', A, \Delta \quad \Gamma', \neg B, \Delta$$

$(\neg\neg)$ rule:

$$\Gamma', \neg\neg A, \Delta$$

$$\mid (\neg\neg)$$

$$\Gamma', A, \Delta$$

Observe that we use the same names for the inference and decomposition rules, as once the we have built the decomposition tree (with use of the decomposition rules) with all leaves being axioms, it constitutes a proof of A in **RS** with branches labeled by the proper inference rules.

Now we still need to introduce few useful definitions and observations.

Indecomposable Sequence

A sequence $\Gamma^{'}$ built only out of literals, i.e. $\Gamma \in \mathcal{F}'^{*}$ is called **an indecomposable sequence**.

Decomposable Formula

A formula that is not a literal is called **a decomposable formula**.

Decomposable Sequence

A sequence Γ that contains a decomposable formula is called **a decomposable sequence**.

Observation 1

For any **decomposable** sequence, i.e. for any $\Gamma \notin \mathcal{F}'^*$ there is **exactly one** decomposition rule that can be applied to it.

This rule is determined by the first decomposable formula in Γ, and by the main connective of that formula.

Observation 2

If the main connective of the first decomposable formula is $\cup, \cap$, or $\Rightarrow$, then the decomposition rule determined by it is $(\cup), (\cap)$, or $(\Rightarrow)$, respectively.

Observation 3

If the main connective of the first decomposable formula is $\neg$, then the decomposition rule determined by it is determined by the *second connective* of the formula. If the second connective is $\cup, \cap, \neg$, or $\Rightarrow$, then corresponding decomposition rule is $(\neg\cup), (\neg\cap), (\neg\neg)$ and $(\neg \Rightarrow)$.

Because of the importance of the above observations we write them in a form of the following

Unique Decomposition Lemma

For any sequence $\Gamma \in \mathcal{F}^*$,

$\Gamma \in \mathcal{F}'^*$ or Γ is in the domain of only one of the **RS** Decomposition Rules.

Decomposition Tree $\mathbf{T}_A$

For each $A \in \mathcal{F}$, a decomposition tree $\mathbf{T}_A$ is a tree build as follows.

Step 1. The formula A is the **root** of $\mathbf{T}_A$ and for any node Γ of the tree we follow the steps below.

Step 2. If Γ is **indecomposable,** then Γ becomes a **leaf** of the tree.

Step 3. If Γ is decomposable, then we traverse Γ from left to right to identify the first **decomposable formula** B and identify the unique (**Unique Decomposition Lemma**) decomposition rule determined by the main connective of B.

We put its left and right premisses as the left and right leaves, respectively.

Step 4. We repeat steps 2 and 3 until we obtain only leaves.

Decomposition Tree $\mathbf{T}_\Gamma$

For each $\Gamma \in \mathcal{F}^*$, a decomposition tree $\mathbf{T}_\Gamma$ is a tree build as follows.

Step 1. The sequence Γ is the **root** of $\mathbf{T}_\Gamma$ and for any node Δ of the tree we follow the steps bellow.

Step 2. If Δ in indecomposable, then Δ becomes a **leaf** of the tree.

Step 3. If Δ is decomposable, then we traverse Δ from left to right to identify the first **decomposable formula** B and identify the unique (**Unique Decomposition Lemma**) decomposition rule determined by the main connective of B.

We put its left and right premisses as the left and right leaves, respectively.

Step 4. We repeat steps 2 and 3 until we obtain only leaves.

Now we prove the following theorem.

Decomposition Tree Theorem

For any sequence $\Gamma \in \mathcal{F}^*$ the following conditions hold.

1. $\mathbf{T}_\Gamma$ is finite and unique.

2. $\mathbf{T}_\Gamma$ is a proof of Γ in **RS** if and only if all its leafs are axioms.

3. $\nvdash_{\mathbf{RS}}$ if and only if $\mathbf{T}_\Gamma$ has a non- axiom leaf.

Proof: The tree $\mathbf{T}_\Gamma$ is unique by the **Unique Decomposition Lemma**. It is finite because there is a finite number of logical connectives in Γ and all decomposition rules diminish the number of connectives. If the tree has a non- axiom leaf it is not a proof by definition. By the its uniqueness it also means that the proof does not exist.

Example

Let's construct, as an example a decomposition tree $\mathbf{T}_A$ of the following formula A.

$$A = ((a \cup b) \Rightarrow \neg a) \cup (\neg a \Rightarrow \neg c))$$

The formula A forms a one element decomposable sequence. The first decomposition rule used is determined by its main connective.

We put a box around it, to make it more visible.

$$((a \cup b) \Rightarrow \neg a) \boxed{\cup} (\neg a \Rightarrow \neg c))$$

The first and only rule applied is $(\cup)$ and we can write the first segment of our *decomposition tree* $\mathbf{T}_A$:

$$((a \cup b) \Rightarrow \neg a) \boxed{\cup} (\neg a \Rightarrow \neg c))$$

$$\mid (\cup)$$

$$((a \cup b) \Rightarrow \neg a), (\neg a \Rightarrow \neg c)$$

Now we decompose the sequence

$$((a \cup b) \Rightarrow \neg a), (\neg a \Rightarrow \neg c).$$

It is a decomposable sequence with the first, decomposable formula

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

The next step of the construction of our decomposition tree is determined by its main connective $\Rightarrow$ (we put the box around it). item[The only rule] determined by the sequence is $(\Rightarrow)$ applied (as decomposition rule) to the sequence

$$((a \cup b) \boxed{\Rightarrow} \neg a), (\neg a \Rightarrow \neg c).$$

The second stage of the decomposition tree is now as follows.

$$((a \cup b) \Rightarrow \neg a) \boxed{\cup} (\neg a \Rightarrow \neg c))$$

$$\mid (\cup)$$

$$((a \cup b) \boxed{\Rightarrow} \neg a), (\neg a \Rightarrow \neg c)$$

$$\mid (\Rightarrow)$$

$$\neg(a \cup b), \neg a, (\neg a \Rightarrow \neg c)$$

The next sequence to decompose is the sequence

$$\neg(a \cup b), \neg a, (\neg a \Rightarrow \neg c)$$

with the first decomposable formula

$$\neg(a \cup b).$$

Its main connective is $\neg$, so to find the appropriate rule we have to examine **next connective**, which is $\cup$.

The decomposition rule determine by this stage of decomposition is $(\neg\cup)$.

Next stage of the construction of the decomposition tree $\mathbf{T}_A$ is as follows.

$$((a \cup b) \Rightarrow \neg a) \boxed{\cup} (\neg a \Rightarrow \neg c))$$

$$\mid (\cup)$$

$$((a \cup b) \boxed{\Rightarrow} \neg a), (\neg a \Rightarrow \neg c)$$

$$\mid (\Rightarrow)$$

$$\boxed{\neg} (a \boxed{\cup} b), \neg a, (\neg a \Rightarrow \neg c)$$

$$\bigwedge (\neg \cup)$$

$$\neg a, \neg a, (\neg a \Rightarrow \neg c) \qquad \neg b, \neg a, (\neg a \Rightarrow \neg c)$$

Now we have two decomposable sequences: $\neg a, \neg a, (\neg a \Rightarrow \neg c)$ and $\neg b, \neg a, (\neg a \Rightarrow \neg c)$.

They both happen to have the same first decomposable formula $(\neg a \Rightarrow \neg c)$. We decompose it and obtain the following:

$$((a \cup b) \Rightarrow \neg a) \boxed{\cup} (\neg a \Rightarrow \neg c))$$

$$\mid (\cup)$$

$$((a \cup b) \boxed{\Rightarrow} \neg a), (\neg a \Rightarrow \neg c)$$

$$\mid (\Rightarrow)$$

$$\boxed{\neg}(a \boxed{\cup} b), \neg a, (\neg a \Rightarrow \neg c)$$

$$\bigwedge (\neg \cup)$$

$$\neg a, \neg a, (\neg a \boxed{\Rightarrow} \neg c) \qquad \neg b, \neg a, (\neg a \boxed{\Rightarrow} \neg c)$$

$$\mid (\Rightarrow) \qquad\qquad \mid (\Rightarrow)$$

$$\neg a, \neg a, \neg\neg a, \neg c \qquad \neg b, \neg a, \neg\neg a, \neg c$$

It is easy to see that we need only one more step to complete the process of constructing the unique decomposition tree of $\mathbf{T}_A$, namely, by decomposing the sequences:

$$\neg a, \neg a, \neg\neg a, \neg c$$

and

$$\neg b, \neg a, \neg\neg a, \neg c.$$

The complete decomposition tree $\mathbf{T}_A$ is:

$$\mathbf{T}_A$$

$$((a \cup b) \Rightarrow \neg a) \boxed{\cup} (\neg a \Rightarrow \neg c))$$

$$\mid (\cup)$$

$$((a \cup b) \boxed{\Rightarrow} \neg a), (\neg a \Rightarrow \neg c)$$

$$\mid (\Rightarrow)$$

$$\boxed{\neg} (a \boxed{\cup} b), \neg a, (\neg a \Rightarrow \neg c)$$

$$\bigwedge (\neg\cup)$$

$$\neg a, \neg a, (\neg a \boxed{\Rightarrow} \neg c) \qquad \neg b, \neg a, (\neg a \boxed{\Rightarrow} \neg c)$$

$$\mid (\Rightarrow) \qquad\qquad \mid (\Rightarrow)$$

$$\neg a, \neg a, \boxed{\neg\neg} a, \neg c \qquad \neg b, \neg a, \boxed{\neg\neg} a, \neg c$$

$$\mid (\neg\neg) \qquad\qquad \mid (\neg\neg)$$

$$\neg a, \neg a, a, \neg c \qquad \neg b, \neg a, a, \neg c$$

All leafs are axioms, the tree represents a proof of A in **RS**

Example Consider now the formula

$$A = (((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$$

and its decomposition tree:

$$\mathbf{T}_A$$

$$(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$$
$$\mid (\cup)$$
$$((a \Rightarrow b) \cap \neg c), (a \Rightarrow c)$$
$$\bigwedge (\cap)$$

$$\begin{array}{cc}
(a \Rightarrow b), (a \Rightarrow c) & \neg c, (a \Rightarrow c) \\
\mid (\Rightarrow) & \mid (\Rightarrow) \\
\neg a, b, (a \Rightarrow c) & \neg c, \neg a, c \\
\mid (\Rightarrow) & \\
\neg a, b, \neg a, c &
\end{array}$$

The above tree $\mathbf{T}_A$ is **unique** by the Decision Tree Theorem and represents the only possible search for proof of the formula

$$A = ((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$$

in **RS**. It has a non-axiom leaf, hence the proof of A in **RS does not exists**; i. e.

$$\nvdash A.$$

We use this information to construct a truth assignment that would falsify the formula A. Such a variable assignment is called a **counter-model generated by the decomposition tree.**

RS: Counter Models Generated by Decomposition Trees
RS: Proof of COMPLETENESS THEOREM

Counter-model generated by the decomposition tree.

Example: Given a formula A:

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

and its decomposition tree $\mathbf{T}_A$ (see next slide).

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

$$\begin{array}{cc}
(a \Rightarrow b), (a \Rightarrow c) & \neg c, (a \Rightarrow c) \\
\mid (\Rightarrow) & \mid (\Rightarrow) \\
\neg a, b, (a \Rightarrow c) & \neg c, \neg a, c \\
\mid (\Rightarrow) & \\
\neg a, b, \neg a, c &
\end{array}$$

Consider a non-axiom leaf:

$$\neg a, b, \neg a, c$$

Let v be any variable assignment

$$v: \ VAR \longrightarrow \{T, F\}$$

such that it makes this non-axiom leaf FALSE, i.e. we put

$$v(a) = T, v(b) = F, v(c) = F.$$

Obviously, we have that

$$v^*(\neg a, b, \neg a, c) = \neg T \cup F \cup \neg T \cup F = F.$$

Moreover, all our rules of inference are **sound** (to be proven formally in the next section).

Rules soundness means that if one of premisses of a rule is FALSE, so is the conclusion.

Hence, the soundness of the rules proves (by induction on the degree of sequences $\Gamma \in \mathbf{T}_A$) that v, as defined above **falsifies all sequences** on the branch of $\mathbf{T}_A$ that ends with the non-axiom leaf $\neg a, b, \neg a, c$.

In particular, the formula A is on this branch, hence

$$v^*(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c)) = F$$

and v is **a counter-model** for A.

The truth assignments defined by a non- axiom leaves are called **counter-models generated** by the decomposition tree.

The construction of the counter-models generated by the decomposition trees are crucial to the proof of the Completeness Theorem for **RS**.

F "climbs" the Tree $\mathbf{T}_A$.

$$\mathbf{T}_A$$

$$(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c)) = \mathbf{F}$$

$$\mid (\cup)$$

$$((a \Rightarrow b) \cap \neg c), (a \Rightarrow c) = \mathbf{F}$$

$$\bigwedge (\cap)$$

$$\begin{array}{cc}
(a \Rightarrow b), (a \Rightarrow c) = \mathbf{F} & \neg c, (a \Rightarrow c) \\
\mid (\Rightarrow) & \mid (\Rightarrow) \\
\neg a, b, (a \Rightarrow c) = \mathbf{F} & \neg c, \neg a, c \\
\mid (\Rightarrow) & \\
\neg a, b, \neg a, c = \mathbf{F} &
\end{array}$$

Observe that the same construction applies to any other non-axiom leaf, if exists, and gives the other "F climbs the tree" picture, and hence other counter- model for A.

By the Uniqueness of the Decomposition Tree Theorem all possible counter-models (restricted) for A are those generated by the non- axioms leaves of the $\mathbf{T}_A$. In our case the formula A has only one non-axiom leaf, and hence only one (restricted) counter model.

RS: COMPLETENESS THEOREM

We prove first the Soundness Theorem for RS; and then the completeness part of the Completeness Theorem.

Soundness Theorem 1

For any $\Gamma \in \mathcal{F}^*$,

if $\vdash_{\mathbf{RS}} \Gamma$, then $\models \Gamma$.

Proof: we prove here as an example the soundness of two of inference rules. We leave the proof for the other rules as an exercise.

We show that rules $(\Rightarrow)$ and $(\neg\cup)$ of **G are sound**.

We show even more, i.e. that the premisses and conclusion of both rules are **logically equivalent**.

I.e. that for all v, $v^*(Premiss(es)) = T$, implies that $v^*(Conclusion) = T$.

We hence show the following.

Equivalency: If $P_1, (P_2)$ are premiss(es) of any rule of RS, C is its conclusion, then $v^*(P_1) = v^*(C)$ in case of one premiss rule and $v^*(P_1) \cap v^*(P_2) = v^*(C)$, in case of the two premisses rule.

Consider the rule $(\cup)$.

$$(\cup) \quad \frac{\Gamma', A, B, \Delta}{\Gamma', (A \cup B), \Delta}.$$

By the definition:

$v^*(\Gamma', A, B, \Delta) = v^*(\delta_{\{\Gamma', A, B, \Delta\}}) = v^*(\Gamma') \cup v^*(A) \cup v^*(B) \cup v^*(\Delta) = v^*(\Gamma') \cup v^*(A \cup B) \cup v^*(\Delta) = v^*(\delta_{\{\Gamma', (A \cup B), \Delta\}}) = v^*(\Gamma', (A \cup B), \Delta)$.

Consider the rule $(\neg\cup)$.

$$(\neg\cup) \quad \frac{\Gamma^{'}, \neg A, \Delta \ : \ \Gamma^{'}, \neg B, \Delta}{\Gamma^{'}, \neg(A \cup B), \Delta}.$$

By the definition:

$v^*(\Gamma^{'}, \neg A, \Delta) \cap v^*(\Gamma^{'}, \neg B, \Delta) = (v^*(\Gamma^{'}) \cup v^*(\neg A) \cup v^*(\Delta)) \cap (v^*(\Gamma^{'}) \cup v^*(\neg B) \cup v^*(\Delta)) = (v^*(\Gamma^{'}, \Delta) \cup v^*(\neg A)) \cap (v^*(\Gamma^{'}, \Delta) \cup v^*(\neg B)) =$ by distributivity $= (v^*(\Gamma^{'}, \Delta) \cup (v^*(\neg A) \cap v^*(\neg B)) = v^*(\Gamma^{'}) \cup v^*(\Delta) \cup (v^*(\neg A \cap \neg B)) =$ by the logical equivalence of $(\neg A \cap \neg B)$ and $\neg(A \cup B) = v^*(\delta_{\{\Gamma', \neg(A \cup B), \Delta\}} = v^*(\Gamma^{'}, \neg(A \cup B), \Delta))$.

Proofs for all other rules follow the above pattern (and proper logical equivalencies).

From the above **Soundness Theorem 1** we get as a corollary, in a case when Γ is a one formula sequence, the following soundness lemma for formulas.

Soundness Theorem 2

For any $A \in \mathcal{F}$,

if $\vdash_{\mathbf{RS}} A$, then $\models A$.

Now we are ready to prove the Completeness Theorem, in two forms: sequence, and formula.

Completeness Theorem 1

For any formula $A \in \mathcal{F}$,

$\vdash_{\mathbf{RS}} A$ if and only if $\models A$.

Completeness Theorem 2

For any $\Gamma \in \mathcal{F}^*$,

$\vdash_{\mathbf{RS}} \Gamma$ if and only if $\models \Gamma$.

Both proofs are carried by proving the contraposition implication to the Completeness Part, as the soundness part has been already proven.

Proof: as an example, we list the main steps in the proof of a contraposition of the **Completeness Theorem 1**.

$$\text{If} \quad \not\vdash_{\mathbf{RS}} A, \quad \text{then} \quad \not\models A.$$

To prove the Completenes Theorem we proceed as follows.

Define, for each $A \in \mathcal{F}$ its decomposition tree (**Decomposition Tree Definition**).

Prove that the decomposition tree is finite unique (**Decomposition Tree Theorem**) and has the following property:

$\vdash_{\mathbf{G}} A$ iff all leafs of the decomposition tree of A are axioms.

What means that if $\not\vdash_{\mathbf{RS}} A$, then there is a leaf L of the decomposition tree of A, which is not an axiom.

Observe, that by soundness, if one premiss of a rule of **RS** is FALSE, so is the conclusion.

Hence by soundness and the definition of the decomposition tree any truth assignment v that falsifies an non axiom leaf, i.e. any v such that $v^*(L) = F$ falsifies A, namely $v^*(A) = F$ and hence constitutes a counter model for A. This ends that proof that $\not\models A$.

Essential part:

Given a formula A such that $\not\vdash_{\mathbf{RS}} A$ and its decomposition tree of A with a non-axiom leaf L.

We define **a counter-model** v determined by the non- axiom leat L as follows:

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

This proves that $\not\models A$ and ends the proof of the **Completeness Theorem** for RS.

Part 2: Gentzen Sequent Calculus GL

The proof system GL for the classical propositional logic 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 like in the original Gentzen system **LK** are Gentzen *sequents*.

Hence we use also a name Gentzen sequent calculus.

Language of GL: $\mathcal{L} = \mathcal{L}_{\{\cup,\cap,\Rightarrow,\neg,\}}$.

We add a new symbol to the alphabet: $\longrightarrow$. It is 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 sign $\longrightarrow$.

We denote, as in the **RS** system, the finite sequences of formulas by Greek capital letters Γ, Δ, Σ, with indices if necessary.

Sequent definition: a sequent is the expression

$$\Gamma \longrightarrow \Delta,$$

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

Meaning of sequents Intuitively, we interpret a sequent

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

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

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

The sequent: $A_1, ..., A_n \longrightarrow$ (where $n \geq 1$) means *that* $A_1 \cap ... \cap A_n$ *yields a contradiction.*

The sequent $\longrightarrow B_1, ..., B_m$ (where $m \geq 1$) means $\models (B_1 \cup ... \cup B_m)$.

The empty sequent: $\longrightarrow$ means *a contradiction.*

Given non empty sequences: Γ, Δ, we denote by

$$\sigma_\Gamma$$

any conjunction of all formulas of Γ, and by

$$\delta_\Delta$$

any disjunction of all formulas of Δ.

The intuitive semantics (meaning, interpretation) of a sequent $\Gamma \longrightarrow \Delta$ (where Γ, Δ are nonempty) is

$$\Gamma \longrightarrow \Delta \;\equiv\; (\sigma_\Gamma \Rightarrow \delta_\Delta).$$

Formal semantics for sequents (expressions of **GL**)

Let $v : VAR \longrightarrow \{T, F\}$ be a truth assignment, v^* its (classical semantics) extension to the set of formulas $\mathcal{F}$.

We extend v^* to the set

$$SEQ = \{\ \Gamma \longrightarrow \Delta :\ \Gamma, \Delta \in \mathcal{F}^*\ \}$$

of all sequents as follows: for any sequent $\Gamma \longrightarrow \Delta \in SEQ$,

$$v^*(\Gamma \longrightarrow \Delta)\ =\ v^*(\sigma_\Gamma) \Rightarrow v^*(\delta_\Delta).$$

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

$$v^*(\ \longrightarrow \Delta)\ =\ (T\ \Rightarrow v^*(\delta_\Delta)),$$

$$v^*(\Gamma \longrightarrow\ )\ =\ (v^*(\sigma_\Gamma) \Rightarrow\ F).$$

The sequent $\Gamma \longrightarrow \Delta$ is *satisfiable* if there is a truth assignment $v : VAR \longrightarrow \{T, F\}$ such that $v^*(\Gamma \longrightarrow \Delta) = T$.

Model for $\Gamma \longrightarrow \Delta$ is any v, such that

$$v^*(\Gamma \longrightarrow \Delta) = T.$$

We write it

$$v \models \Gamma \longrightarrow \Delta$$

Counter- model is any v such that

$$v^*(\Gamma \longrightarrow \Delta) = F.$$

We write it

$$v \not\models \Gamma \longrightarrow \Delta.$$

Tautology is any sequent $\Gamma \longrightarrow \Delta$, such that $v^*(\Gamma \longrightarrow \Delta) = T$ for all truth assignments $v : VAR \longrightarrow \{T, F\}$, i.e.

$$\models \Gamma \longrightarrow \Delta.$$

Example Let $\Gamma \longrightarrow \Delta$ be a sequent

$$a, (b \cap a) \longrightarrow \neg b, (b \Rightarrow a).$$

The truth assignment v for which $v(a) = T$ and $v(b) = T$ is a model for $\Gamma \longrightarrow \Delta$, as shows the following computation.

$v^*(a, (b \cap a) \longrightarrow \neg b, (b \Rightarrow a)) = v^*(\sigma_{\{a,(b\cap a)\}}) \Rightarrow v^*(\delta_{\{\neg b,(b\Rightarrow a)\}}) = v(a) \cap (v(b) \cap v(a)) \Rightarrow \neg v(b) \cup (v(b) \Rightarrow v(a)) = T \cap T capT \Rightarrow \neg T \cup (T \Rightarrow T) = T \Rightarrow (F \cup T) = T \Rightarrow T = T.$

Observe that the only v for which $v^*(\Gamma) = v^*(a, (b \cap a) = T$ is the above $v(a) = T$ and $v(b) = T$ that is a model for $\Gamma \longrightarrow \Delta$.

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

$$\models \ a, (b \cap a) \ \longrightarrow \ \neg b, (b \Rightarrow a).$$

Definition of GL

Axioms of GL: Any sequent of variables (positive literals) which contains a propositional variable that appears on both sides of the sequent arrow $\longrightarrow$, i.e any sequent of the form

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

for any $a \in VAR$ and any sequences $\Gamma'_1, \Gamma'_2, \Delta'_1, \Delta'_2 \in VAR^*$.

Inference rules of GL

We denote by $\Gamma^{'}$, $\Delta^{'}$ finite sequences formed out of literals i.e. out of propositional variables or negations of propositional variables. Γ, Δ denote any finite sequences of formulas.

Conjunction rules

$$(\cap \rightarrow)\ \frac{\Gamma', A, B, \Gamma \ \longrightarrow\ \Delta'}{\Gamma', (A \cap B), \Gamma \ \longrightarrow\ \Delta'},$$

$$(\rightarrow \cap)\ \frac{\Gamma \ \longrightarrow\ \Delta, A, \Delta' \ ; \ \Gamma \ \longrightarrow\ \Delta, B, \Delta'}{\Gamma \ \longrightarrow\ \Delta, (A \cap B), \Delta'},$$

Disjunction rules

$$(\rightarrow \cup)\ \frac{\Gamma \ \longrightarrow\ \Delta, A, B, \Delta'}{\Gamma \ \longrightarrow\ \Delta, (A \cup B), \Delta'},$$

$$(\cup \rightarrow)\ \frac{\Gamma', A, \Gamma \ \longrightarrow\ \Delta' \ ; \ \Gamma', B, \Gamma \ \longrightarrow\ \Delta'}{\Gamma', (A \cup B), \Gamma \ \longrightarrow\ \Delta'},$$

Implication rules

$$(\to\Rightarrow)\ \frac{\Gamma', A, \Gamma \ \longrightarrow\ \Delta, B, \Delta'}{\Gamma', \Gamma \ \longrightarrow\ \Delta, (A \Rightarrow B), \Delta'},$$

$$(\Rightarrow\to)\ \frac{\Gamma', \Gamma \ \longrightarrow\ \Delta, A, \Delta' \ ;\ \Gamma', B, \Gamma \ \longrightarrow\ \Delta, \Delta'}{\Gamma', (A \Rightarrow B), \Gamma \ \longrightarrow\ \Delta, \Delta'},$$

Negation rules

$$(\neg\to)\ \frac{\Gamma', \Gamma \ \longrightarrow\ \Delta, A, \Delta'}{\Gamma', \neg A, \Gamma \ \longrightarrow\ \Delta, \Delta'},$$

$$(\to\neg)\ \frac{\Gamma', A, \Gamma \ \longrightarrow\ \Delta, \Delta'}{\Gamma', \Gamma \ \longrightarrow\ \Delta, \neg A, \Delta'}.$$

We define:

$$\mathbf{GL} = (SEQ, AL, (\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg \Rightarrow), (\neg\neg))$$

Formal proof of a sequent $\Gamma \longrightarrow \Delta$ in the proof system **GL** we understand any sequence

$$\Gamma_1 \longrightarrow \Delta_1,\ \Gamma_2 \longrightarrow \Delta_2,\ ....,\ \Gamma_n \longrightarrow \Delta_n$$

of sequents of formulas (elements of SEQ), such that $\Gamma_1 \longrightarrow \Delta_1 \in AL$, $\Gamma_n \longrightarrow \Delta_n = \Gamma \longrightarrow \Delta$, and for all i $(1 < i \leq n)$ $\Gamma_i \longrightarrow \Delta_i \in AL$, or $\Gamma_i \longrightarrow \Delta_i$ is a conclusion of one of the inference rules of **GL** with all its premisses placed in the sequence $\Gamma_1 \longrightarrow \Delta_1,\ ....\Gamma_{i-1} \longrightarrow \Delta_{i-1}$.

We write, as usual,

$$\vdash_{\mathbf{GL}} \Gamma \longrightarrow \Delta$$

to denote that $\Gamma \longrightarrow \Delta$ has a formal proof in **GL**.

A formula $A \in \mathcal{F}$, has a proof in if the sequent $\longrightarrow A$ has a proof in **GL**, i.e. we define:

$$\vdash_{\mathbf{GL}} A \quad iff \quad \longrightarrow \; A.$$

A proof tree, or **GL**-proof of $\Gamma \longrightarrow \Delta$ is a tree

$$\mathbf{T}_{\Gamma \longrightarrow \Delta}$$

of sequents satisfying the following conditions:

1. The topmost sequent, i.e *the root* of $\mathbf{T}_{\Gamma \longrightarrow \Delta}$ is $\Gamma \longrightarrow \Delta$,

2. *All leafs* are axioms,

3. *The nodes* are sequents such that each sequent on the tree follows from the ones immediately preceding it by one of the rules.

We write the proof- trees indicating additionally the name of the inference rule used at each step of the proof.

Remark Proof search, i.e. **decomposition tree** for a given formula A and hence a **proof** of A in **GL** is not always unique!!

For example, a tree-proof (in **GL**) of the de Morgan law

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

is the following.

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

$$\mid (\longrightarrow \Rightarrow)$$

$$\neg(a \cap b) \; \longrightarrow \; (\neg a \cup \neg b)$$

$$\mid (\longrightarrow \cup)$$

$$\neg(a \cap b) \; \longrightarrow \; \neg a, \neg b$$

$$\mid (\longrightarrow \neg)$$

$$b, \neg(a \cap b) \; \longrightarrow \; \neg a$$

$$\mid (\longrightarrow \neg)$$

$$b, a, \neg(a \cap b) \; \longrightarrow$$

$$\mid (\neg \longrightarrow)$$

$$b, a \longrightarrow \; (a \cap b)$$

$$\bigwedge (\longrightarrow \cap)$$

$$b, a \longrightarrow \; a \qquad\qquad b, a \longrightarrow \; b$$

Exercise 1 : Write all other proofs of $\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$ in **GL**.

Exercise 2: Verify that the axiom and the rules of inference of **GL** are sound, i.e. that the following theorem holds.

Soundness Theorem for GL: For any sequent $\Gamma \longrightarrow \Delta \in SEQ$,

$$if \ \ \vdash_{\mathbf{GL}} \Gamma \longrightarrow \Delta \ \ then \ \ \ \models \ \Gamma \longrightarrow \Delta.$$

Completeness Theorem For any sequent $\Gamma \longrightarrow \Delta \in SEQ$,

$$\vdash_{\mathbf{GL}} \Gamma \longrightarrow \Delta \ \ iff \ \ \ \models \ \Gamma \longrightarrow \Delta.$$

The proof of the Completeness Theorem is similar to the proof for the **RS** system and is assigned as an exercise.