# CHAPTER 11

# Two Gentzen Style Proof Systems for Classical Logic

Hilbert style systems are easy to define and admit a simple proof of the Completeness Theorem but they are difficult to use. By humans, not mentioning computers. Their emphasis is on logical axioms, keeping the rules of inference at a minimum.

Gentzen systems reverse this situation by emphasizing the importance of inference rules, reducing the role of logical axioms to an absolute minimum. They may be 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. He dealt with a complicated structure, called *sequents*. We present here a version (without structural rules) of his original formalization. His exact formalizations for classical logic and for intuitionistic logic are presented in the next chapter.

The other automated proof system presented here is due to H. Rasiowa and R. Sikorski and appeared for the first time in 1961. It is inspired by Gentzen original system and is equivalent to it (like may others), so all of them are called *Gentzen style proof systems*.

The Rasiowa and Sikorski system (RS System) is more elegant and easier to understand, so we present it first.

## 1 The Gentzen Style System RS

### Language

Let $\mathcal{F}$ denote a set of formulas of $\mathcal{L} = \mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}}$. The rules of inference of our system **RS** will operate on *finite sequences of formulas*, i.e. elements of $\mathcal{F}^*$, instead of just plain formulas $\mathcal{F}$, as in Hilbert style formalizations. It means that we adopt as the set of expressions $\mathcal{E}$ of **RS** the set $\mathcal{F}^*$, i.e

$$\mathcal{E} = \mathcal{F}^*.$$

We will denote the finite sequences of formulas 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 $\Gamma$ true.

As we know, the disjunction in classical logic is associative and commutative, i.e., for any formulas $A, B, C \in \mathcal{F}$, the formulas $A \cup (B \cup C)$, $(A \cup B) \cup C$, $A \cup (C \cup B)$, $(B \cup A) \cup C$, $C \cup (B \cup A)$, $C \cup (A \cup B)$, $(C \cup A) \cup B$, etc... are logically equivalent. We adopt hence a notation

$$\delta_{\{A,B,C\}} = A \cup B \cup C$$

to denote any disjunction of formulas $A, B, C$.

In a general case, for any sequence $\Gamma \in \mathcal{F}^*$, if $\Gamma$ is of a form

$$A_1, A_2, ..., A_n \tag{1}$$

then by $\delta_\Gamma$ we will understand any disjunction of all formulas of $\Gamma$, i.e.

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

**Formal Semantics for RS**

Let $v : VAR \longrightarrow \{T, F\}$ be a truth assignment, $v^*$ its 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. For any sequence $\Gamma \in \mathcal{F}^*$, if $\Gamma$ is of the form 1, then we define:

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

**Model**

The sequence $\Gamma$ 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 is also called *a model* for $\Gamma$.

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

**Tautology**

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

**Example**

Let $\Gamma$ 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 $\Gamma$, as shows the following computation.

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

**Example**

Let $\Gamma$ be a sequence

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

and let $v$ be a truth assignment for which $v(a) = T$. Directly from the the definition, we get that

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

This proves that $v$ is a *model* for $\Gamma$ and $\Gamma$ is *satisfiable*.

Assume now that $\Gamma$ is *falsifiable* i.e. that we have a truth assignment $v$ for which

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

This is possible only when (in short-hand notation)

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

what is impossible as $(\neg b \cup b) = T$ for all $v$. This contradiction proves that $\Gamma$ is *a tautology*.

**Axioms and Rules of Inference**

The rules of inference of **RS** are of the form:

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

where $\Gamma_1, \Gamma_2$ and $\Gamma$ are sequences $\Gamma_1, \Gamma_2$ are called premisses and $\Gamma$ 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 denote the rule that introduces the logical connective $\circ$ in the conclusion sequent $\Gamma$ by $(\circ)$. The notation $(\neg\circ)$ means that the negation of the logical connective $\circ$ is introduced in the conclusion sequence $\Gamma$.

As our language contains the connectives: $\cap, \cup, \Rightarrow$ and $\neg$, so we we are going to define seven inference rules: $(\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg \Rightarrow)$, and $(\neg\neg)$.

We denote, as usual, the set of all axioms by $AX$.

In order to define the axioms $AX$ and the set of rules of inference of **RS** we need to introduce some definitions.

**Literals**

We form a special subset $\mathcal{F}^{'} \subseteq \mathcal{F}$ of formulas, called a set of all *literals*, which is defined as follows.

$$\mathcal{F}^{'} = VAR \cup \{\neg a : a \in VAR\}.$$

The variables are called *positive literals* and the elements of the second set of the above union are called *negative literals*. I.e propositional variables are called positive literals and the negation of a variable is called a negative literal, a variable or a negation of propositional variable is called a literal.

**Indecomposable formulas**

Literals are also called the indecomposable formulas.

Now we form *finite sequences* out of formulas (and, as a special case, out of literals). We need to distinguish the sequences formed out of literals from the sequences formed out of other formulas, so we adopt the following notation.

We denote by $\Gamma^{'}$, $\Delta^{'}$, $\Sigma^{'}$ finite sequences (empty included) formed out of *literals* i.e. out of the elements of $\mathcal{F}^{'}$ i.e. we assume that $\Gamma^{'}$, $\Delta^{'}$, $\Sigma^{'} \in \mathcal{F}^{'*} \subseteq \mathcal{F}^{*}$.

We will denote by $\Gamma, \Delta, \Sigma$ the elements of $\mathcal{F}^{*}$ i.e the finite sequences (empty included) formed out of elements of $\mathcal{F}$.

**Axioms of RS**

As the *axiom* of **RS** we adopt any sequence of literals which contains any propositional variable and its negation, i.e any sequence of the form

$$\Gamma_1^{'}, a, \Gamma_2^{'}, \neg a, \Gamma_3^{'} \tag{2}$$

or of the form

$$\Gamma_1^{'}, \neg a, \Gamma_2^{'}, a, \Gamma_3^{'} \tag{3}$$

for any variable $a \in VAR$ and any sequences $\Gamma_1^{'}, \Gamma_2^{'}, \Gamma_3^{'} \in \mathcal{F}'^{*}$.

**Inference rules of RS**

We define formally the inference rules of **RS** as follows.

**Disjunction rules**

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

**Conjunction rules**

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

**Implication rules**

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

**Negation rule**

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

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

**The Proof System RS**

Formally we define:

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

where $(\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg\Rightarrow), (\neg\neg)$ are the inference rules defined above and $AX$ is the axiom of the system defined by the schemas 2 and 3.

We define the notion of a *formal proof* in **RS** as in any proof system, i.e., by a formal proof of a sequence $\Gamma$ in the proof system $\mathbf{RS} = (\mathcal{F}^*, AX, (\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg\Rightarrow), (\neg\neg))$ we understand any sequence

$$\Gamma_1\Gamma_2....\Gamma_n$$

of sequences of formulas (elements of $\mathcal{F}^*$, such that $\Gamma_1 \in AX$, $\Gamma_n = \Gamma$, and for all i $(1 < i \leq n)$ $\Gamma_i \in AL$, or $\Gamma_i$ is a conclusion of one of the inference rules of **RS** with all its premisses placed in the sequence $\Gamma_1\Gamma_2....\Gamma_{i-1}$.

As the proof system under consideration is fixed, we will write, as usual,

$$\vdash \Gamma$$

instead of $\vdash_{\mathbf{RS}} \Gamma$ to denote that $\Gamma$ has a formal proof in **RS**.

As the proofs in **RS** are sequences (definition of the formal proof) of sequences of formulas (definition of **RS** ) we will not use "," to separate the steps of the proof, or will write sometimes the formal proof (i. e. the sequence $\Gamma_1\ \Gamma_2\ ....\ \Gamma_n$) in a vertical form

$$\begin{array}{c} \Gamma_1 \\ \Gamma_2 \\ ... \\ \Gamma_n. \end{array}$$

We write, however, the formal proofs in **RS** in a form of trees rather then in a form of sequences, ie. in a form of a tree, where *leafs* of the tree are axioms, *nodes* are sequences such that each sequence on the tree follows from the ones immediately preceding it by one of the rules. The *root* is a theorem. We picture, and write our tree-proofs 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 adopt hence the following definition.

**Proof Tree**

By a proof tree, or **RS**-proof of $\Gamma$ 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 $\Gamma$,

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.

In particular cases 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 given formula $A$ from 3 *axioms* was obtained by the subsequent use of the rules $(\cap), (\cup), (\cup), (\cap), (\cup), (\neg\neg)$, and $(\Rightarrow)$, we represent it as the following proof tree:

$$A\ (\textit{conclusion of}\ (\Rightarrow))$$

$$
\begin{array}{c}
| \, (\Rightarrow) \\
conclusion\ of\ (\neg\neg) \\
| \, (\neg\neg) \\
conclusion\ of\ (\cup) \\
| \, (\cup) \\
conclusion\ of\ (\cap) \\
\bigwedge(\cap)
\end{array}
$$

$$
\begin{array}{cc}
conclusion\ of\ (\cap) & conclusion\ of\ (\cup) \\
| \, (\cup) & | \, (\cup) \\
axiom & conclusion\ of\ (\cap) \\
 & \bigwedge(\cap) \\
 & \\
 & axiom \quad axiom
\end{array}
$$

The proof trees are often called *derivation trees* and we will use this notion as well. Remark that the derivation trees don't represent a different *definition* of a formal proof. 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.

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

$$
\begin{array}{cc}
a, (\neg a \cup \neg b) & b, (\neg a \cup \neg b) \\
| \, (\cup) & | \, (\cup) \\
a, \neg a, \neg b & b, \neg a, \neg b
\end{array}
$$

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. The formal proof (with comments) thus obtained will be the following sequence of elements of $\mathcal{F}^*$:

$$a, \neg a, \neg b \quad (axiom)$$
$$b, \neg a, \neg b \quad (axiom)$$
$$a, (\neg a \cup \neg b) \quad (rule(\cup))$$
$$b, (\neg a \cup \neg b) \quad (rule(\cup))$$
$$(a \cap b), (\neg a \cup \neg b) \quad (rule(\cap))$$
$$\neg\neg(a \cap b), (\neg a \cup \neg b) \quad (rule(\neg\neg))$$
$$(\neg(a \cap b) \Rightarrow (\neg a \cup \neg b)) \quad (rule(\Rightarrow)).$$

# 2 Search for Proofs and Decomposition Trees

The main advantage of our system and the the Gentzen style systems in general lies not in a way we generate proofs in them, but in the way we can *search* for proofs in them. That happens to be deterministic and automatic.

Before we describe the general proof search procedure, let us consider few examples.

**Example**

Consider now a formula $A$ of the form of another de Morgan law

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

Obviously it should have a proof in **RS**. The search for it consists of building a certain tree, called a **decomposition tree**. We proceed as follows.

Observe that the *main connective* of $A$ is $\Rightarrow$. So, if it *had* a proof in **RS** it would have come from the *only possible rule* used in its last step, namely the rule $(\Rightarrow)$ applied to a sequence $\neg\neg(a \cup b), (\neg a \cap \neg b)$. So the last step in the proof of $A$ would look as follows.

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

Now, if the sequence $\neg\neg(a \cup b), (\neg a \cap \neg b)$ (and hence also had our formula ) had a proof in **RS** its *only step* at this stage would have been the application of the rule $(\neg\neg)$ to a sequence $(a \cup b), (\neg a \cap \neg b)$. So, if A had a proof, its last two steps would have been:

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

Again, if the sequence $(a \cup b), (\neg a \cap \neg b)$ had a proof in **RS** its *only step* at this stage would have been the application of the rule $(\cup)$ to a sequence $a, b, (\neg a \cap \neg b)$. So, if A had a proof, its last three steps would have been as follows.

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

Now, if the sequence $a, b, (\neg a \cap \neg b)$ had a proof in **RS** its *only step* at this stage would have been the application of the rule $(\cap)$ to the sequences $a, b, \neg a$ and $a, b, \neg b$ as its left and right premisses, respectively. Both sequences are axioms and the following tree is a proof of $A$ in **RS**.

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

$$
\begin{array}{c}
| \ (\neg\neg) \\
(a \cup b), (\neg a \cap \neg b) \\
| \ (\cup) \\
a, b, (\neg a \cap \neg b) \\
\bigwedge (\cap) \\
\\
a, b, \neg a \qquad\qquad\qquad a, b, \neg b
\end{array}
$$

From the above proof tree of $A$ we construct, if we want, 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. I.e. the following sequence of elements of $\mathcal{F}^*$ is a *formal proof* of $(\neg(a \cup b) \Rightarrow (\neg a \cap \neg b))$ in **RS**.

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

**Example**

Given a formula A of the form

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

Observe that the *main connective* of A is $\cup$. So, if it *had* a proof in **RS** it would have come from the *only possible rule* used in its last step, namely the rule $(\cup)$ applied to a sequence $((a \Rightarrow b) \cap \neg c), (a \Rightarrow c)$. So the last step in the proof of A would have been:

$$
\begin{array}{c}
(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c)) \\
| \ (\cup) \\
((a \Rightarrow b) \cap \neg c), (a \Rightarrow c)
\end{array}
$$

Now, if the sequence $((a \Rightarrow b) \cap \neg c), (a \Rightarrow c)$ (and hence also had our formula ) had a proof in **RS** its *only step* at this stage would have been the application of the rule $(\cap)$ to the sequences $(a \Rightarrow b), (a \Rightarrow c)$ and $\neg c, (a \Rightarrow c)$ as its left and right premisses, respectively. So, if A had a proof, its last two steps would have been:

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

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

Now, if the sequences $(a \Rightarrow b), (a \Rightarrow c)$ and $\neg c, (a \Rightarrow c)$ had proofs in **RS**, then their last, and the only steps would have been the the separate application of the rule $(\Rightarrow)$ to the sequences $\neg a, b, (a \Rightarrow c)$ and $\neg c, \neg a, c$, respectively. The sequence $\neg c, \neg a, c$ is an axiom, so we stop the search on this branch. The sequence $\neg a, b, (a \Rightarrow c)$ is not an axiom, so the search continues. In this case we can go one step further: if $\neg a, b, (a \Rightarrow c)$ had a proof it would have been only by the application of the rule $(\Rightarrow)$ to a sequence $\neg a, b, \neg a, \ c$ which is not an axiom and the search ends.

The tree generated by this search is called **a decomposition tree** and is the following.

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

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

As the tree generated by this search contains a non-axiom leaf and hence is **not a proof**.

## 2.1 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 generates its *unique, finite* decomposition tree, $\mathbf{T}_A$ such that if all its leafs are axioms, the tree constitutes the proof of $A$ in **RS**. If there is a leaf of $\mathbf{T}_A$ 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 $\Gamma', A, B, \Delta$ 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 a proof search decomposition 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**

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

**Conjunction decomposition rules**

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

**Implication decomposition rules**

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

**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:**

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

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

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

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

$$\Gamma^{'}, (A \cap B), \Delta$$
$$\bigwedge(\cap)$$
$$\Gamma^{'}, A, \Delta \qquad\qquad \Gamma^{'}, B, \Delta$$

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

$$\Gamma^{'}, \neg(A \cap B), \Delta$$
$$| \ (\neg\cap)$$
$$\Gamma^{'}, \neg A, \neg B, \Delta$$

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

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

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

$$\Gamma^{'}, \neg(A \Rightarrow B), \Delta$$
$$\bigwedge(\neg \Rightarrow)$$

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

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

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

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.

A sequence $\Gamma^{'}$ built only out of literals, i.e. $\Gamma \in \mathcal{F}'^{*}$ is called **an indecomposable sequence**. A formula that is not a literal is called **a decomposable formula**. A sequence $\Gamma$ 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 $\Gamma$, 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

**Lemma 2.1 (Unique Decomposition)** *For any sequence* $\Gamma \in \mathcal{F}^*$*,*

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

### 2.1.1 Decomposition Tree Definition

**Decomposition Tree $\mathbf{T}_A$**

For each formula $A \in \mathcal{F}$, 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 $\Gamma$ of the tree we follow the steps below.

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

**Step 3.** If $\Gamma$ is decomposable, then we traverse $\Gamma$ from left to right to identify the first **decomposable formula** $B$ and identify the unique (Lemma 2.1) 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 $\Gamma$ is the **root** of $\mathbf{T}_\Gamma$ and for any node $\Delta$ of the tree we follow the steps bellow.
**Step 2.** If $\Delta$ in indecomposable, then $\Delta$ becomes a **leaf** of the tree.
**Step 3.** If $\Delta$ is decomposable, then we traverse $\Delta$ from left to right to identify the first **decomposable formula** $B$ and identify the unique (Lemma 2.1) 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.

### 2.1.2 Decomposition Tree Properties

We prove now the basic properties of the Decomposition Tree. They are essential to the proof of the Completeness Theorem for RS¿

**Theorem 2.1** *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* $\Gamma$ *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 Lemma 2.1. It is finite because there is a finite number of logical connectives in $\Gamma$ 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. 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), hence the only rule determined by the sequence is $(\Rightarrow)$. 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 rule determine by this stage of decomposition is $(\neg\cup)$ and now the next stage of the decomposition tree $\mathbf{T}_A$ is as follows.

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

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

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:

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

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

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:

$$\mathbf{T}_A$$

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

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

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:

$$\mathbf{T}_A$$

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

$$| \, (\cup)$$

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

$$\bigwedge (\cap)$$

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

$$| \, (\Rightarrow) \qquad | \, (\Rightarrow)$$

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

$$| \, (\Rightarrow)$$

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

The above tree $\mathbf{T}_A$ is unique by the theorem 2.1 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.

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.

**Counter-model generated by the decomposition tree**

Given a formula $A = ((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$ and its decomposition tree $\mathbf{T}_A$.

Consider a non-axiom leaf $\neg a, b, \neg a, c$ of $\mathbf{T}_A$. Let $v$ be any variable assignment $v : \ VAR \ \longrightarrow \ \{T, F\}$ such that

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

Obviously,

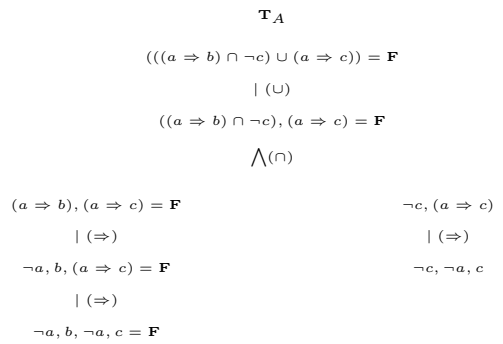
$$v^*(\neg a, b, \neg a, c) = \neg T \cup F \cup \neg T \cup F = F.$$

Moreover, all our rules of inference are sound (to be proven formally in the next section). It means that if one of premisses of a rule is **false**, so is the conclusion. This proves 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 variable assignments constructed in this way are called the **counter-model generated** by the decomposition tree. We represent this graphically as follows.

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




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 2.1 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.

**Exercises**

Write 10 formulas, build their decomposition trees and find for each case when there is a non-axiom leaf a counter-model determined by the tree.

# 3 Completeness of RS

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

**Theorem 3.1 (Soundness Theorem 1)** *For any* $\Gamma \in \mathcal{F}^*$,

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

We prove here as an example the soundness of two of inference rules ($\Rightarrow$) and ($\neg\cup$). We leave the proof for the other rules as an exercise.

We show even more, 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 ($\neg\cup$).

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

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 of RS follow the above pattern (and proper logical equivalencies).

From the above Soundness Theorem 1 3.1 we get as a corollary, in a case when $\Gamma$ is a one formula sequence, the following soundness Theorem for formulas.

**Theorem 3.2 (Soundness Theorem 2)**

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

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

**Theorem 3.3 (Completeness Theorem 1)**

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

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

**Theorem 3.4 (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**.

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

To prove this we proceed as follows.

**We define,** for each $A \in \mathcal{F}$ its *decomposition tree* (section 2.1.1).

**Prove** that the decomposition tree is finite and *unique* (theorem 2.1) and has the following property:

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

**What means** that if $\nvdash_{\mathbf{RS}} A$, 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 $\nvdash_{\mathbf{RS}} A$ 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.

# 4 Gentzen Sequent Calculus GL

We present here a proof system **GL** for the classical propositional logic that is a version (without structural rules) of the original Gentzen systems **LK**.

We give a constructive of the completeness theorem for the system **GL**. The proof is very similar to the proof of the completeness theorem for the system **RS**.

The axioms, the rules of inference of the system **GL** operate, as in the original system **LK**, on expressions called by Gentzen *sequents*, hence the name Gentzen sequent calculus.

**Language of GL**

We adopt a propositional language $\mathcal{L} = \mathcal{L}_{\{\cup,\cap,\Rightarrow,\neg\}}$ with the set of formulas denoted by $\mathcal{F}$ and we add a new symbol

$$\longrightarrow$$

called a Gentzen arrow, to it. As the next step we build expressions called *sequents*. 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 $\Gamma, \Delta, \Sigma$, with indices if necessary. We define a sequent as follows.

**Sequent**

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

$$\Gamma \longrightarrow \Delta$$

is called a sequent. $\Gamma$ is called the **antecedent** of the sequent, $\Delta$ is called the **succedent**, and each formula in $\Gamma$ and $\Delta$ is called a **sequent-formula**.

We denote the sequents by the letter $S$, with or without subscripts.

**Semantics of sequents**

Intuitively, a sequent $A_1, ..., A_n \longrightarrow B_1, ..., B_m$ (where $n, m \geq 1$) means: *if* $A_1 \cap ... \cap A_n$ *then* $B_1 \cup ... \cup B_m$.

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 *that* $B_1 \cup ... \cup B_m$ *is true.*

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

Given non empty sequences $\Gamma$, $\Delta$, we denote by

$$\sigma_\Gamma$$

*any conjunction* of all formulas of $\Gamma$, and by

$$\delta_\Delta$$

*any disjunction* of all formulas of $\Delta$.

The intuitive semantics for a sequent $\Gamma \longrightarrow \Delta$ (where $\Gamma, \Delta$ are nonempty) is hence that it is logically equivalent to the formula $(\sigma_\Gamma \Rightarrow \delta_\Delta)$, i.e.

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

**Formal semantics for sequents**

Formally, let $v : VAR \longrightarrow \{T, F\}$ be a Boolean truth assignment, $v^*$ its 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:

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

**Model**

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

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

to denote that $v$ is a model for $\Gamma \longrightarrow \Delta$.

**Counter- model**

The sequent $\Gamma \longrightarrow \Delta$ is *falsifiable* if there is a truth assignment $v$, such that $v^*(\Gamma \longrightarrow \Delta) = F$. In this case $v$ is called a *counter-model* for $\Gamma \longrightarrow \Delta$ and we write it as

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

**Tautology**

The sequent $\Gamma \longrightarrow \Delta$ is a *tautology* if $v^*(\Gamma \longrightarrow \Delta) = T$ for all truth assignments $v : VAR \longrightarrow \{T, F\}$ and we write

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

to denote that $\Gamma \longrightarrow \Delta$ is a tautology.

**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.

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

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 $\Gamma \longrightarrow \Delta$. Hence it is impossible to find $v$ which would falsify it, what proves that $\Gamma \longrightarrow \Delta$ is a tautology, i.e.

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

**The Proof System GL**

The rules of inference of **GL** are of the form:

$$\frac{S_1}{S} \quad or \quad \frac{S_1 \ ; \ S_2}{S},$$

where $S_1, S_2$ and $S$ are sequents. $S_1, S_2$ are called premisses and $S$ is called the conclusion of the rule of inference.

Each rule of inference introduces a new logical connective to the antecedent or to the succedent of the conclusion sequent.

We denote the rule that introduces the logical connective $\circ$ to the antecedent of the conclusion sequent $S$ by $(\circ \rightarrow)$. The notation $(\rightarrow \circ)$ means that the logical connective is introduced to the succedent of the conclusion sequent $S$.

As our language contains the connectives: $\cap, \cup, \Rightarrow$ and $\neg$, we are going to adopt the following inference rules: $(\cap \rightarrow)$ and $(\rightarrow \cap)$, $(\cup \rightarrow)$ and $(\rightarrow \cup)$, $(\Rightarrow\rightarrow)$ and $(\rightarrow\Rightarrow)$, and finally, $(\neg \rightarrow)$ and $(\rightarrow \neg)$.

We denote by $\Gamma^{'}$, $\Delta^{'}$ finite sequences formed out of **positive literals** i.e. out of propositional variables only.

They are also called **indecomposable sequents**.

$\Gamma$, $\Delta$ denote any finite sequences of formulas.

**Axioms of GL**

As the axioms of **GL** we adopt any indecomposable sequent sequent which contains a positive literal a (variable) that appears on both sides of the sequent arrow $\longrightarrow$, i.e any sequent of the form

$$\Gamma'_1, a, \Gamma'_2 \ \longrightarrow \ \Delta'_1, a, \Delta'_2, \tag{4}$$

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

**Inference rules of GL**

The inference rules of **GL** are defined as follows.

**Conjunction rules**

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

**Disjunction rules**

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

**Implication rules**

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

**Negation rules**

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

Formally we define:

$$\mathbf{GL} = (\mathcal{L}, SQ, AX, (\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg\Rightarrow), (\neg\neg))$$

where $SQ = \{\ \Gamma \longrightarrow \Delta :\ \Gamma, \Delta \in \mathcal{F}^*\ \}$, $(\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg\Rightarrow), (\neg\neg)$ are the inference rules defined above and $AX$ is the axiom of the system defined by the schema 4.

We define the notion of a *formal proof* in **GL** as in any proof system, i.e., by a 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 $\Gamma_1 \longrightarrow \Delta_1 \in AX$, $\Gamma_n \longrightarrow \Delta_n = \Gamma \longrightarrow \Delta$, 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**, or we write simply

$$\vdash \Gamma \longrightarrow \Delta$$

when the system **GL** is fixed.

We say that a formula $A \in \mathcal{F}$, has a proof in **GL** and denote it by $\vdash_{\mathbf{GL}} A$ if the sequent $\longrightarrow A$ has a proof in **GL**, i.e. we define:

$$\vdash_{\mathbf{GL}} A \quad iff \quad \vdash_{\mathbf{GL}} \ \longrightarrow\ A. \tag{5}$$

We write, however, the formal proofs in **GL** in a form of proof trees rather then in a form of sequences.

**Proof trees**

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. the *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 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.

In particular cases we will write our proof- trees indicating additionally the name of the inference rule used at each step of the proof.

For example, a proof-tree (in **GL**) of the de Morgan law $(\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$ is the following.

$$\begin{array}{c} \longrightarrow\ (\neg(a \cap b) \Rightarrow (\neg a \cup \neg b)) \\ | \ (\longrightarrow\Rightarrow) \\ \neg(a \cap b)\ \longrightarrow\ (\neg a \cup \neg b) \\ | \ (\longrightarrow \cup) \\ \neg(a \cap b)\ \longrightarrow\ \neg a, \neg b \\ | \ (\longrightarrow \neg) \\ b, \neg(a \cap b)\ \longrightarrow\ \neg a \\ | \ (\longrightarrow \neg) \\ b, a, \neg(a \cap b)\ \longrightarrow \\ | \ (\neg \longrightarrow) \\ b, a \longrightarrow\ (a \cap b) \\ \bigwedge(\longrightarrow \cap) \end{array}$$

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

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

**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.

**Theorem 4.1 (Soundness of GL)**

*For any sequent* $\Gamma \longrightarrow \Delta \in SEQ$,

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

In particular, following the definition 5 we get the following.

**Theorem 4.2 (Formula Soundness of GL)**

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

$$if \ \vdash_{\mathbf{GL}} A \ \ then \ \models A.$$

There are the following two completeness theorems for **GL**. The second, called formula completeness theorem is a particular case of the first theorem.

**Theorem 4.3 (Completeness Theorem)**

*For any sequent* $\Gamma \longrightarrow \Delta \in SEQ$,

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

Following the definition 5 we get, as a particular case of the above the following theorem.

**Theorem 4.4 (Formula Completeness Theorem)**

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

$$\vdash_{\mathbf{GL}} A \ \ iff \ \models A.$$

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