CHAPTER 6

ch6

Propositional Automated Proof Systems

1 Gentzen Style Proof System RS

sec:rs

Hilbert style systems are easy to define and admit different proofs 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, with obligatory Modus Ponens, at a minimum.

Gentzen style proof 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 allow us to define effective automatic procedures for proof search, what was impossible in a case of the Hilbert style systems. For this reason they are also called *automated proof systems.* They serve as formal models of computing systems that automate the reasoning process. Building computing systems means providing an algorithmic description to a formal proof system so that it can be implemented on a computer to prove theorems in an efficient manner.

The first proof systems of this style was invented by G. Gentzen in 1934, hence the name. His proof systems for classical and intuitionistic predicate logics introduced special expressions built of formulas called *sequents.* Hence the Gentzen style systems using sequents as basic expressions are often called sequent systems, or Gentzen sequent systems, or simply Gentzen formalizations.

We present here (section 4) two Gentzen systems **GL** and **G** for classical propositional logic and prove their completeness. We also present a propositional version of Gentzen original system **LK** and discuss a proof of Gentzen Hauptsatz for it. Hauptsatz is literally rendered as the *main theorem* and is known as *cut-elimination theorem.* We prove the equivalency of the cut-free propositional **LK** and the complete system **G**. The Gentzen original formalization for intuitionistic propositional logic **LI** is discussed and presented in chapter **??**. The classical and intuitionistic predicate versions are discussed in chapter **??**.

The other historically important automated proof system is due to Rasiowa and Sikorski (1960). Their proof systems for classical propositional and predicate logic use as basic expressions sequences of formulas, less complicated then Gentzen sequents. As they were inspired Gentzen systems we call them, as we call many others similarly inspired, Gentzen style proof system, or Gentzen style formalization. The Rasiowa and Sikorski proof system is simpler and easier to understand then the Gentzen sequent systems. Hence their system **RS** is the first to be presented here in section 1.

Historical importance and lasting influence of Rasiowa and Sikorski work lays in the fact that they were first to use the proof searching capacity of their proof system to define a constructive method of proving the completeness theorem for both propositional and predicate classical logic. We introduce and explain in detail their method and use it prove the completeness of the **RS** in section 2.2. We also introduce and discuss two other **RS** style system **RS1** and **RS2** in in section 3. We also generalize the **RS** completeness proof method to the Gentzen sequent systems and prove the completeness of **GL** and **G** systems in section 4.1. The completeness proof for proof system **RSQ** for classical predicate logic is presented in chapter **??**.

2 Proof System RS

We present here a propositional version of the original Rasiowa and Sikorski (1960) Gentzen style proof system for classical logic. We call it **RS system** for **R**asiowa-**S**ikorski. The **RS** system extends naturally to predicate logic **QRS** system which is presented in chapter **??**. Both systems admit a constructive proof of Completeness Theorem. We prove completeness of **RS** in section 2.2. We define components and semantics of the system **RS** as follows.

Components of the proof system RS

Language $\mathcal{L}$

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** operate on *finite sequences of formulas*, i.e. elements of $\mathcal{F}^*$, unlike on plain formulas $\mathcal{F}$ in Hilbert style formalizations.

Expressions $\mathcal{E}$

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 expressions of **RS**, i.e. the finite sequences of formulas by Γ, Δ, Σ, with indices if necessary.

Semantic Link

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. 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. In particular we write

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

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

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

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

gamma

then by δ_Γ we will understand any disjunction of all formulas of Γ. We write it informally

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

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 $|calE$ of expressions of **RS** , i.e. to the set $\mathcal{F}^*$ of all finite sequences of $\mathcal{F}$ as follows. For any sequence $\Gamma \in \mathcal{F}^*$, if Γ is the sequence (1), then we define:

$$v^*(\Gamma) = v^*(\delta_\Gamma). \tag{2}$$

s-gamma

Model

A 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 is called a **model** for Γ. We denote it as

$$v \models \Gamma. \tag{3}$$

m-gamma

Counter- Model

A sequence Γ is said to be **falsifiable** if there is a truth assignment v, such that $v^*(\Gamma) = F$. Such a truth assignment is] called a **counter-model** for Γ. We write it symbolically as

$$v \not\models \Gamma. \tag{4}$$

cm-gamma

Tautology

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

$$\models \Gamma. \tag{5}$$

t-gamma

Exercise 1

Let Γ *be a sequence* $a, (b \cap a), \neg b, (b \Rightarrow a)$.

1. Show that the truth assignment $v : VAR \longrightarrow \{T, F\}$*, such that* $v(a) = F$ *and* $v(b) = T$ *falsifies* Γ*, i.e.* $v \not\models \Gamma$.

2. Let Γ *be a sequence* $a, (\neg b \cap a), \neg b, (a \cup b)$ *and let* v *be a truth assignment for which* $v(a) = T$*. Prove that* $v \models \Gamma$.

3. Let Γ *be a sequence* $a, (\neg b \cap a), \neg b, (a \cup b)$*. Prove that* $\models \Gamma$.

Solution

1. Γ is the sequence $a, (b \cap a), \neg b, (b \Rightarrow a)$. We eveluate $v^*(\Gamma) = v^*(\delta_\Gamma) = v^*(a) \cup v^*(b \cap a) \cup v^*(\neg b) \cup v^*(b \Rightarrow a) = F \cup (F \cap T) \cup F \cup (T \Rightarrow F) = F \cup F \cup F \cup F = F$. By (4) we proved $v \not\models \Gamma$.

2. Let Γ be a sequence $a, (\neg b \cap a), \neg b, (a \cup b)$. We eveluate $v^*(\Gamma) = v^*(\delta_\Gamma) = v^*(a) \cup v^*(\neg b \cap a) \cup v^*(\neg b) \cup v^*(a \cup b) = T \cup v^*(\neg b \cap a) \cup v^*(\neg b) \cup v^*(a \cup b) = T$. By (3) we proved $v \models \Gamma$.

3. Assume now that Γ 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 Γ that (5) holds and Γ is a **tautology**.

In order to define the axioms LA 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.

$$LT = VAR \cup \{\neg a : a \in VAR\}. \quad (6)$$ `literal`

The variables are called **positive literals** and the elements of the second set of the above union (6) 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 and sequences

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

$$\Gamma^{'},\ \Delta^{'},\ \Sigma^{'},\ldots \quad \text{with indices if necessary,} \tag{7}$$ indecomp

elements of $LT^{*} \subseteq \mathcal{F}^{*}$, i.e. $\Gamma^{'},\ \Delta^{'},\ \Sigma^{'}$ are finite sequences (empty included) formed out of **literals**. We call them **indecomposable sequences**.

We denote by

$$\Gamma,\ \Delta,\ \Sigma,\ldots \quad \text{with indices if necessary,} \tag{8}$$ seq

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

Logical Axioms LA

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

$$\Gamma^{'}_{1}, a, \Gamma^{'}_{2}, \neg a, \Gamma^{'}_{3} \tag{9}$$ axiom1

or of the form

$$\Gamma^{'}_{1}, \neg a, \Gamma^{'}_{2}, a, \Gamma^{'}_{3} \tag{10}$$ axiom2

for any variable $a \in VAR$ and any sequences $\Gamma^{'}_{1}, \Gamma^{'}_{2}, \Gamma^{'}_{3} \in LT^{*}$ of literals.

Semantic Link
Consider axiom (9). Directly from the extension of the notion of tautology to **bf RS** (5), we have that for any truth assignments $v : VAR \longrightarrow \{T, F\}$, $v^{*}(\Gamma^{'}_{1}, \neg a, \Gamma^{'}_{2}, a, \Gamma^{'}_{3}) = v^{*}(\Gamma^{'}_{1}) \cup v^{*}(\neg a) \cup v^{*}(a) \cup v^{*}(\Gamma^{'}_{2}, \Gamma^{'}_{3}) = v^{*}(\Gamma^{'}_{1}) \cup T \cup v^{*}(\Gamma^{'}_{2}, \Gamma^{'}_{3}) = T$ The same applies to the axiom (10) We have thus proved the following.

t-axioms **Fact 1**

Logical axioms of **RS** *are tautologies.*

Rules of inference $\mathcal{R}$

All rules of inference of **RS** are of the form

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

where $\Gamma_1,\ \Gamma_2,\ \Gamma \in \mathcal{F}^{*}$, i.e. $\Gamma_1,\ \Gamma_2,\ \Gamma$ are any finite sequences (**??**) of formulas. The sequences Γ_1, Γ_2 are called **premisses** and Γ is called a **conclusion** of the rule of inference.

Each rule of inference of **RS** introduces a new logical connective, or a negation of a logical connective. We denote a rule of inerence 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. As our language contains the connectives: $\cap, \cup, \Rightarrow$ and $\neg$, so we we are going to define the following seven inference rules:

$$(\cup),\ (\neg\cup),\ (\cap),\ (\neg\cap),\ (\Rightarrow),\ (\neg\ \Rightarrow),\ \text{and}\ (\neg\ \neg). \qquad (11)$$ `rules`

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

The Proof System RS

Formally we define the proof system **RS** as follows.

$$\mathbf{RS} = (\mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}},\ \mathcal{E},\ \ LA,\ \ \mathcal{R}), \qquad (12)$$ `def:rs`

where $\mathcal{E} = \{\Gamma : \ \ \Gamma \in \mathcal{F}^*\}$, LA contains logical axioms of the system defined by the schemas (9) and (10), $\mathcal{R}$ is the set of rules of inference:

$$\mathcal{R} = \{(\cup),\ (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg \Rightarrow), (\neg\neg)\}$$

defined by (11).

By a **formal proof** of a sequence Γ in the proof system **RS** we understand any sequence

$$\Gamma_1,\ \Gamma_2, ...\ \Gamma_n \tag{13}$$ `fp`

of sequences of formulas (elements of $\mathcal{F}^*$, such that

$\Gamma_1 \in LA$, $\Gamma_n = \Gamma$,

and for all i $(1 \leq i \leq n)$ $\Gamma_i \in AL$, or Γ_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 Γ 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, and write the formal proof as $\Gamma_1;\ \Gamma_2;\ ....\ \Gamma_n$.

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.

Definition 1 (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.

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 tree-proof of a given formula A from *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:

$$A \ (conclusion\ of\ (\Rightarrow))$$

$$\mid (\Rightarrow)$$

$$conclusion\ of\ (\neg\neg)$$

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

$$conclusion\ of\ (\cup)$$

$$\mid (\cup)$$

$$conclusion\ of\ (\cap)$$

$$\bigwedge(\cap)$$

$$conclusion\ of\ (\cap) \qquad\qquad conclusion\ of\ (\cup)$$

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

$$axiom \qquad\qquad conclusion\ of\ (\cap)$$

$$\bigwedge(\cap)$$

$$axiom \qquad axiom$$

The proof trees are often called **derivation trees** and we will use this notion as well. Remark that the proof trees don't represent a different *definition* of a formal proof. Trees represent a certain *visualization* of the proofs and any formal proof in any system can be represented in a tree form.

Example 1

Here is a proof tree in **RS** *of the de Morgan law* $(\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$.

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

$$\mid (\Rightarrow)$$

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

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

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

$$\bigwedge(\cap)$$

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

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

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

To obtain a "linear " formal proof of $(\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$ 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 is:

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

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

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

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

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

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

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

2.1 Search for Proofs and Decomposition Trees

The main advantage of the Gentzen style proof systems lies not in a way we generate proofs in them, but in the way we can *search* for proofs in them. That such proof searches happens to be deterministic and automatic. Before we describe a general proof search procedure for **RS** let us look at few simple examples. 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** as we want it to be, and will prove later to be complete. The search for the proof consists of building a certain tree. We call it a **decomposition tree**, to be defined formally later. We proceed as follows.

Observe that the *main connective* of A is $\Rightarrow$. So, if A *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 its premiss, namely 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.

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

$$\mid (\Rightarrow)$$

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

Now, if the sequence $\neg\neg(a \cup b), (\neg a \cap \neg b)$ (and hence also the formula A) 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:

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

$$\mid (\Rightarrow)$$

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

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

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

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.

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

$$\mid (\Rightarrow)$$

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

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

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

$$\mid (\cup)$$

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

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**.

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

$$\mid (\Rightarrow)$$

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

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

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

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

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**.

$$a, b, \neg b$$

$$a, b, \neg a$$

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

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

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

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

Consider now 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 A *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:

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

$$\mid (\cup)$$

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

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:

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

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

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.

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

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

The tree generated by this search contains a **non-axiom leaf**, so by definition, it **is not a proof**.

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

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

Tree Decomposition Rules

$(\cup)$ **rule**

$$\begin{array}{c}\Gamma^{'}, (A \cup B), \Delta \\ \mid (\cup) \\ \Gamma^{'}, A,\ B,\ \Delta\end{array}$$

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

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

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

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

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

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

$$\bigwedge(\cap)$$

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

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

$$\begin{array}{c}\Gamma^{'}, \neg(A \cap B), \Delta \\ \mid (\neg \cap) \\ \Gamma^{'}, \neg A, \neg B, \Delta\end{array}$$

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

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

$$\mid (\cup)$$

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

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

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

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

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

$(\neg \neg)$ **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.

indecomp **Definition 2**

1. A sequence Γ is **indecomposable** *if and only if $\Gamma \in LT^*$.*

2. A formula A is **decomposable** *if and only if $A \in \mathcal{F} - LT$.*

3. A sequence Γ is **decomposable** *if and only if it contains a decomposable formula.*

Directly from the definition 8 we have three simple, but important observations.

observ **Fact 2**

1. For any decomposable sequence Γ, i.e. for any $\Gamma \notin LT^*$ 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.*

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.

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 $(\neg \Rightarrow)$.

Directly from the Fact 2 we we have the following lemma.

unique **Lemma 1 (Unique Decomposition)**

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

$\Gamma \in LT^*$ *or* Γ *is in the domain of only one of the* **RS** *Decomposition Rules.*

Now we define formally, for any formula $A \in \mathcal{F}$ and $\Gamma \in \mathcal{F}^*$ their decompositions trees. The decomposition tree for for the formula A is a particular case (one element sequence) of the tree for a sequence Γ.

def:dtree **Definition 3 (Decomposition Tree $\mathbf{T}_A$)**

For each formula $A \in \mathcal{F}$*, its 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* **decomposition rule** *determined by the main connective of* B*. In case of a one premisses rule we put is premise as a leaf; in case of a two premisses rule we put its left and right premisses as the left and right leaves respectively.*

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

We now prove the following Decomposition Tree Theorem 1. This Theorem provides a crucial step in the proof of the Completeness Theorem for **RS**.

dtree **Theorem 1 (Decomposition Tree)**

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 1. 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**.

Exercise 2

Construct a decomposition tree $\mathbf{T}_A$ *of the following formula A.*

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

Solution
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* $\mathbf{T}_A$:

$$\mathbf{T}_A$$

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

$$\mid (\cup)$$

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

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.

$$\mathbf{T}_A$$

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

$$\mid (\cup)$$

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

$$\mid (\Rightarrow)$$

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

The next sequence to decompose is the sequence $\neg(a \cup b), \neg a, (\neg a \Rightarrow \neg c)$.
The first decomposable formula is $\neg(a \cup b)$. Its main connective is $\neg$, so determine the appropriate decomposition 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.

$$\mathbf{T}_A$$

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

$$\mid (\cup)$$

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

$$\mid (\Rightarrow)$$

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

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

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

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 simultenously and obtain the following:

$$\mathbf{T}_A$$

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

$$\mid (\cup)$$

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

$$\mid (\Rightarrow)$$

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

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

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

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

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

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:

$$\mathbf{T}_A$$

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

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

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

Exercise 3

Prove that the formula $A = (((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$ *is not provable in* **RS***, i.e.*

$$\nvdash_{\mathbf{RS}} (((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c)).$$

Solution
We construct the formula A decomposition tree as follows.

$$\mathbf{T}_A$$

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

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

The above tree $\mathbf{T}_A$ is unique by the Theorem 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 by Theorem 1 the proof of A in **RS** does not exists.

rscompl

2.2 Strong Soundness and Completeness

Our main goal is to prove the Completeness Theorem for **RS**. The proof of completeness presented here is due to Rasiowa and Sikorski, as is the proof system **RS**. Their proof, and the proof system was inverted for the classical predicate logic and was published in 1961. We present their predicate logic proof system **QRS** together with the proof of its completeness in chapter **??**. Both completeness proofs, for propositional **RS** and predicate **QRS** proof systems, are constructive as they are based on a direct construction of a counter model for any unprovable formula. The construction of a counter model for a formula A uses directly its decomposition tree $\mathbf{T}_A$. We call such constructed model a *counter model determined* by the tree $\mathbf{T}_A$. Both proofs relay heavily of the notion of a *strong soundness.* We define it now, adopting Chapter **??** general definition to our semantics.

r-strsound

Definition 4 (Strongly Sound Rules)

Given a proof system $S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R})$ An inference rule $r \in \mathcal{R}$ of the form

$$(r) \qquad \frac{P_1 \ ; \ P_2 \ ; \ .... \ ; \ P_m}{C}$$

is **strongly sound** *(undef classical semantics) if the following condition holds for all $v : VAR \longrightarrow \{T, F\}$*

$$v \ \models \{P_1, P_2, .P_m\} \quad \text{if and only if} \quad v \ \models \ C. \tag{14}$$ ssound

We say it less formally that a rule (r) is **strongly sound** *if the conjunction of its premisses is logically equivalent with the conclusion, i.e.*

$$P_1 \cap P_2 \cap \ \dots \ \cap P_m \equiv C. \tag{15}$$ ss-equiv

S-strsound

Definition 5 (Strongly Sound S)

A proof system $S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R})$ is strongly sound (undef classical semantics) if and only if all logical axioms LA are tautologies and all its rules of inference $r \in \mathcal{R}$ are **strongly sound**.

thm:ssound

Theorem 2 (Strong Soundness)

The proof system **RS** *(18) is* **strongly sound**.

Proof

The logical axioms (9), (10) are tautologies by Fact **??**. We prove as an example the **strong soundness** of two of inference rules: $(\cup)$ and $(\neg\cup)$. Proofs for all other rules follow the same patterns and are left as an exercise. By definition 4 of strong soundness we have to show the condition (15). Written formally it says that we have to show that that if P_1, P_2 are premisses of a given rule and C is its conclusion, then for all truth assignments $v : VAR \longrightarrow \{T, F\}$, $v^*(P_1) = v^*(C)$ in case of one premiss rule, and $v^*(P_1) \cap v^*(P_2) = v^*(C)$, in case of a 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 $\neg(A \cup B) = v^*(\delta_{\{\Gamma', \neg(A \cup B), \Delta\}} = v^*(\Gamma^{'}, \neg(A \cup B), \Delta))$.

Observe that the strong soundness implies soundness (not only by name!), hence we have also proved the following.

thm:sound

Theorem 3 (Soundness for RS)

For any $\Gamma \in \mathcal{F}^*$,
if $\vdash_{\boldsymbol{RS}} \Gamma$, *then* $\models \Gamma$. *In particular, for any* $A \in \mathcal{F}$, *if* $\vdash_{\boldsymbol{RS}} A$, *then* $\models A$.

We have just proved (Theorem 2) that all the rules of inference of **RS** of are strongly sound, i.e. $C \equiv P$ and $C \equiv P_1 \cap P_2$. The strong soundness of the rules means that if *at least* one of premisses of a rule is *false*, so is its conclusion. Hence given a formula A, such that its $\mathbf{T}_A$ has a branch ending with a non-axiom leaf. By Strong Soundness Theorem 2, any v that make this non-axiom leaf false also falsifies all sequences on that branch, and hence falsifies the formula A. This means that any v, such that it falsifies a non-axiom leaf is a **counter-model** for A. We have hence proved the following.

thm:c-model

Theorem 4 (Counter Model)

Given a formula $A \in \mathcal{F}$ such that its decomposition tree $\mathbf{T}_A$ contains a **non-axiom** *leaf L_A. Any truth assignment v that falsifies the non-axiom leaf L_A is a counter model for A. We call it a* **counter-model** *for A* **determined** *by the decomposition tree $\mathbf{T}_A$.*

Here is a simple example explaining how the construction of a counter-model determined by the decomposition tree of a works. Consider a tree

$$\mathbf{T}_A$$

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

$$\mid (\cup)$$

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

$$\bigwedge(\cap)$$

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

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

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

$$\mid (\Rightarrow)$$

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

The tree $\mathbf{T}_A$ has a non-axiom leaf $L_A: \ \neg a, \ b, \neg a, \ c$. The truth assignment $v: \ VAR \longrightarrow \{T, F\}$ that falsifies the leaf $\neg a, \ b, \neg a, \ c$ must be such that

$v^*(\neg a, b, \neg a, c) = v^*(\neg a) \cup v^*(b) \cup v^*(\neg a) \cup v^*(c) = \neg v(a) \cup v(b) \cup \neg v(a) \cup v(c) = F$, i.e. v must be such that $\neg v(a) \cup v(b) \cup \neg v(a) \cup v(c) = F$. We hence get that $v(a) = T, \ \ v(b) = F, \ \ \ v(c) = F$. By the Counter Model Theorem 4, the truth assignment v determined by the non-axiom leaf also falsifies the formula A, i.e. we proved that v is a counter model for A and

$$\not\models (((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c)).$$

The Counter Model Theorem 4, says that the logical value **F** determined by the evaluation a non-axiom leaf "climbs" the decomposition tree. We picture it as follows.

$$\mathbf{T}_A$$

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

$$\mid (\cup)$$

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

$$\bigwedge(\cap)$$

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

Observe that the same counter model construction applies to any other non-axiom leaf of $\mathbf{T}_A$, if exists. The other non-axiom leaf of $\mathbf{T}_A$ defines another evaluation of the non- axiom leaf to $\mathbf{F}$ that also "climbs the tree" and hence defines another counter- model for a formula A. By Counter Model 4 all possible restricted counter-models for A are those determined by its all non- axioms leaves.

In our case the tree $\mathbf{T}_A$ has only one non-axiom leaf, and hence the formula $(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$ only only one restricted counter model.

Our main goal is to prove the Completeness Theorem for **RS**. We prove first the Completeness Theorem for formulas $A \in \mathcal{F}$ and then we generalize it to any sequences $\Gamma \in \mathcal{F}^*$.

thm:compl

Theorem 5 (Completeness Theorem)

For any formula $A \in \mathcal{F}$,
1. $\vdash_{\mathbf{RS}} A$ if and only if $\models A$, and for any $\Gamma \in \mathcal{F}^*$,
2. $\vdash_{\mathbf{RS}} \Gamma$ if and only if $\models \Gamma$.

Proof
Case 1. We have already proved the Soundness Theorem 3, so we need to prove only the completeness part of it, namely to prove the implication:

$$\text{if } \models A, \text{ then } \vdash_{\mathbf{RS}} A. \tag{16}$$ 1

We prove instead of the opposite implication:

$$\text{if } \not\vdash_{\mathbf{RS}} A \text{ then } \not\models A. \tag{17}$$ 2

Assume that A is any formula is such that $\not\vdash_{\mathbf{RS}} A$. By the Decomposition Tree Theorem 1 the tree $\mathbf{T}_A$ contains a non-axiom leaf L_A. We use the non-axiom leaf L_A **to define** a truth assignment $v : VAR \longrightarrow \{T, F\}$ which **falsifies** it as follows:

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

By the Counter Model Theorem 4we have that v also **falsifies** the formula A. We proved that

$$\not\models A$$

and it ends the proof of the case 1.

Case 2. Assume that $\Gamma \in \mathcal{F}^*$ is any sequence such that $\not\vdash_{\mathbf{RS}} \Gamma$. But obviously, $\vdash_{\mathbf{RS}} \Gamma$ if and only if $\vdash_{\mathbf{RS}} \delta_\Gamma$, where δ_Γ is any disjunction of all formulas of Γ. So $\not\vdash_{\mathbf{RS}} \Gamma$ if and only if $\not\vdash_{\mathbf{RS}} \delta_\Gamma$ and by already proven Case 1, $\not\models \ \delta_\Gamma$ what is obviously equivalent to $\not\models \ \Gamma$. This ends the proof of Case 2 and Completeness Theorem.

3 Proof Systems RS1 and RS2

sec:rs1

We present here a two modifications of the system **RS** as an exercise of importance of paying close attention to the syntax. Proof systems might be, as all presented here **RS** type systems are, semantically identical, nevertheless they are very different proof systems.

Language of **RS1** is the same as the language of **RS**, i.e.

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

Rules of inference of **RS1** operate as rules of **RS** on **finite sequences** of formulas and we adopt

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

as the set of expressions of **RS1** . We denote them, as before, by $\Gamma, \Delta, \Sigma \ldots$, with indices if necessary.

The proof system **RS1** contains seven inference rules, denoted by the same symbols as the rules of **RS**, namely $(\cup)$, $(\neg\cup)$, $(\cap)$, $(\neg\cap)$, $(\Rightarrow)$, $(\neg \Rightarrow)$, $(\neg\neg)$.

The inference rules of **RS1** are quite similar to the rules of **RS** Look at them carefully to see where lies the difference.

Reminder Any propositional variable, or a negation of propositional variable is called a **literal**. The set $LT = VAR \cup \{\neg a : \quad a \in VAR\}$ is called a set of all propositional literals. The variables are called positive literals. Negations of variables are called negative literals. We denote, as before, by $\Gamma^{'},\ \Delta^{'},\ \Sigma^{'}, \ldots$ finite sequences (empty included) formed out of literals. We adopt all logical axiom of **RS** as the axioms of **RS1**, i.e. logical axioms LA of **RS1** are:

$$\Gamma^{'}_1,\ a,\ \Gamma^{'}_2,\ \neg a,\ \Gamma^{'}_3,$$

$$\Gamma^{'}_1,\ \neg a,\ \Gamma^{'}_2,\ a,\ \Gamma^{'}_3$$

where $a \in VAR$ is any propositional variable.

We define the inference rules of **RS1** 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 $\Gamma\in\mathcal{F}^*,\ \Delta^{'}\in LT^*,\ A,\ B\ \in\mathcal{F}$.

Proof System RS1

Formally we define the proof system **RS1** as follows.

$$\mathbf{RS1}=(\mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}},\ \mathcal{E},\ \ LA,\ \ \mathcal{R}), \tag{18}$$ `def:rs`

where $\mathcal{E}=\{\Gamma:\ \ \Gamma\in\mathcal{F}^*\}$, LA is the set logical axioms and $\mathcal{R}$ is the set of rules of inference defined above.

Exercise 4

Construct a proof in **RS1** *of a formula*

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

Solution
The decomposition tree below is a **proof** of A in **RS1** as all its leaves are axioms.

$$\mathbf{T}_A$$

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

$$\mid(\Rightarrow)$$

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

$$| (\cup)$$

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

$$| (\neg\neg)$$

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

$$\bigwedge(\cap)$$

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

Exercise 5

Prove that **RS1** *is strongly sound.*

Solution
Observe that the system **RS1** is obtained from **RS** by changing the sequence $\Gamma^{'}$ into Γ and the sequence Δ into $\Delta^{'}$ in all of the rules of inference of **RS**. These changes do not influence the essence of proof of **strong soundness** of the rules of **RS**. One has just to replace the sequence $\Gamma^{'}$ by Γ and the sequence Δ by $\Delta^{'}$ in the proof of strong soundness of each rule of **RS** to obtain a corresponding proof of strong soundness of corresponding rule of **RS1**. We do it, for example for the rule $(\cup)$ of **RS1**. Consider the rule $(\cup)$ of **RS1**:

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

We evaluate:

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

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

Exercise 6

Define in your own words, for any formula $A \in \mathcal{F}$ *the decomposition tree* $\mathbf{T}_A$ *in* **RS1**.

Solution
The definition of the decomposition tree $\mathbf{T}_A$ is again, it its essence similar to the one for **RS** except for the changes which reflect the differences in the corresponding rules of inference. We follow now the following steps.
Step 1 Decompose A using a rule defined by its main connective.
Step 2 Traverse resulting sequence Γ on the new node of the tree from **right** to **left** and find the first decomposable formula.
Step 3 Repeat **Step 1** and **Step 2** until there is no more decomposable formulas. **End** of tree construction.

Exercise 7

Prove the following **Completeness Theorem** *for* **RS1**.

c-rs1 **Theorem 6**

For any formula $A \in \mathcal{F}$,
1. $\vdash_{\mathbf{RS1}} A$ *if and only if* $\models A$, *and for any* $\Gamma \in \mathcal{F}^*$,
2. $\vdash_{\mathbf{RS1}} \Gamma$ *if and only if* $\models \Gamma$.

Solution Part 1.
Observe that directly from the definition of the uniqueness of the decomposition tree $\mathbf{T}_A$ we have that the following holds.

fact **Fact 3**

The decomposition tree $\mathbf{T}_A$ *is a* **proof** *if and only if all leaves are axioms and the proof does not exist otherwise, i.e. we have that* $\nvdash_{\mathbf{RS1}} A$ *if and only if there is a non- axiom leaf on* $\mathbf{T}_A$.

The Fact 3 together with strong soundness of the rules of inference of **RS1** justify the correctness of construction of a counter-model generated by a the a non- axiom leaf and hence the correctness of the following **proof** of the Completeness Theorem.

We prove, as we did in case of **RS** the implication

$$\text{if} \quad \nvdash_{\mathbf{RS1}} A \;\; \text{then} \quad \not\models A.$$

Assume that A is any formula such that $\nvdash_{\mathbf{RS1}} A$. By the Fact 3 the decomposition tree $\mathbf{T}_A$ contains a non-axiom leaf L_A. We use the non-axiom leaf L_A and **define** a truth assignment v which falsifies A, as follows:

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

This proves, by the strong soundness of **RS1**, that $\not\models A$.

The proof of Part 2. is identical to the proof in **RS** case.

Proof System RS2 (19) RS2

System **RS2** is a proof system obtained from **RS** by changing the sequences Γ' into Γ in **all of the rules** of inference of **RS**. The **logical axioms** LA remain the same. Observe that now the decomposition tree may not be unique

Exercise 8

Construct ***two*** *decomposition trees in* **RS2** *of the formula*

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

Solution
Here are two out of many more decomposition trees.

$$\mathbf{T1}_A$$

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

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

$a, a.\neg a, (\neg a \cup \neg b)$	$a, a, (\neg a \cup \neg b)$	$a, \neg b, \neg a$	$a, \neg b, (\neg a \cup \neg b)$
$\mid (\cup)$	$\mid (\cup)$	*axiom*	$\mid (\cup)$
$a, a.\neg a, \neg a, \neg b$	$a, a, \neg a, \neg b$		$a, \neg b, \neg a, \neg b$
axiom	*axiom*		*axiom*

The other tree is:

$$\mathbf{T2}_A$$

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

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

Exercise 9

Explain why the system **RS2** *is* **strongly sound**. *You can use the strong soundness of the system* **RS**.

Solution
The only one difference between **RS** and **RS2** is that in **RS2** each inference rule has at the beginning a sequence of any formulas, not only of literals, as in **RS**. So there are many ways to apply the *decomposition rules* while constructing the decomposition tree, but it does not affect strong soundness, since for all rules of **RS2** premisses and conclusions are still logically equivalent as they were in **RS** .

Consider, for example, **RS2** rule

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

We evaluate $v^*(\Gamma, A, B, \Delta) = v^*(\Gamma) \cup v^*(A) \cup v^*(B) \cup v^*(\Delta) = v^*(\Gamma) \cup v^*(A \cup B) \cup v^*(\Delta) = v^*(\Gamma, (A \cup B), \Delta)$. Similarly, as in **RS**, we show all other rules of **RS2** to be strongly sound, thus **RS2** is sound.

Exercise 10

Define shortly, in your own words, for any formula A, its decomposition tree $\mathbf{T}_A$ *in* **RS2**. *Justify why your definition is correct. Show that in* **RS2** *the decomposition tree for some formula A may not be unique.*

Solution
Given a formula A. The decomposition tree $\mathbf{T}_A$ can be defined as follows. It has

A as a **root**. For each **node**, if there is a rule of **RS2** which conclusion has the same form as node sequence, i.e. there is a decomposition rule to be applied, then the node has children that are premises of the rule. If the **node** consists only of literals (i.e. there is no decomposition rule to be applied), then it does not have any children. The last statement define a termination condition for the tree $\mathbf{T}_A$.

This definition defines correctly the decomposition tree $\mathbf{T}_A$ as it identifies and uses appropriate decomposition rules. Since all rules of inference of **RS2** have a sequence Γ instead of Γ' as it was in **RS**, the choice of the decomposition rule for a node may not unique. For example consider a node $(a => b), (b \cup a)$. Γ in the **RS2** rules may be a sequence of formulas, not only literals, so for the node $(a => b), (b \cup a)$ we can choose as a decomposition rule either $(=>)$ or $(\cup)$. This leads to a non-unique tree.

Exercise 11

Prove the following Completeness Theorem for **RS2**.

c-rs2 **Theorem 7**

For any formula $A \in \mathcal{F}$,
1. $\vdash_{\mathbf{RS2}} A$ *if and only if* $\models A$, *and for any* $\Gamma \in \mathcal{F}^*$,
2. $\vdash_{\mathbf{RS2}} \Gamma$ *if and only if* $\models \Gamma$.

Solution
We need to prove the completeness part only, as the Soundness has been already proved, i.e. we have to prove the implication (Part 1): for any formula A,

$$\text{if } \nvdash_{RS2} A \text{ then } \not\models A.$$

Assume $\nvdash_{RS2} A$. Then **every** decomposition tree of A has at least one non-axiom leaf. Otherwise, there would exist a tree with all axiom leaves and it would be a proof for A. Let $\mathcal{T}_A$ be a set of all decomposition trees of A. We choose an arbitrary $T_A \in \mathcal{T}_A$ with at least one non-axiom leaf L_A. We use the non-axiom leaf L_A to define a truth assignment v which falsifies A, as follows:

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

The value for a sequence that corresponds to the leaf in is F. Since, because of the strong soundness F "climbs" the tree, we found a counter-model for A. This proves that $\not\models A$. Part 2. proof is identical to the proof in **RS** case.

Exercise 12

Write a procedure $TREE_A$ such that for any formula A of **RS2** *it produces its* **unique** *decomposition tree.*

Solution
Here is the procedure.

Procedure $TREE_A$(Formula A, Tree T)
{
$B = ChoseLeftMostFormula(A)$ // Choose the left most formula that is not a literal
$c = MainConnective(B)$ // Find the main connective of B
$R = FindRule(c)$// Find the rule which conclusion that has this connective
$P = Premises(R)$// Get the premises for this rule
$AddToTree(A, P)$// add premises as children of A to the tree
For all p in P // go through all premises
$TREE_A(p, T)$ // build subtrees for each premiss
}

Exercise 13

Prove ***completeness*** *of your Procedure $TREE_A$.*

Solution
Procedure $TREE_A$ provides a unique tree, since it always chooses the most left indecomposable formula for a choice of a decomposition rule and there is only one such rule. This procedure is equivalent to **RS** system, since with the decomposition rules of **RS** the most left decomposable formula is always chosen. The proof **RS** system is complete, thus this Procedure $TREE_A$ is **complete**.

4 Gentzen Sequent Systems GL, G, LK

sec:Gentzen

Gentzen proof systems **GL** and **G** for the classical propositional logic presented here are inspired by and all are versions of the original (1934) Gentzen system **LK**. Their axioms, the rules of inference of the proof system considered here operate, as the original Gentzen system **LK**, on expressions called by Gentzen *sequents*, hence the name Gentzen sequent proof systems, called also Gentzen sequent calculus, or sequents calculus. The original system **LK** is presented and discussed in detail in section 6.

4.1 Gentzen Sequent Systems GL and G

sec:GL

The system **GL** presented here is the most similar in its structure to the system **RS** (18) and hence is the first to be considered. It admits a constructive proof of the Completeness Theorem that is very similar to the proof of the Completeness Theorem for the system **RS**.

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. It means we consider formally a new language $\mathcal{L}_1 = \mathcal{L}_{\{\cup,\cap,\Rightarrow,\neg\}} \cup \{\longrightarrow\}$. As the next step we build expressions called *sequents* out of $\mathcal{L}_1$. The sequents are built out of finite sequences (empty included) of formulas, i.e. elements of $\mathcal{F}^*$ of $\mathcal{L}_{\{\cup,\cap,\Rightarrow,\neg\}}$, and the additional sign $\longrightarrow$.

We denote , as in the **RS** system, the finite sequences of formulas of of $\mathcal{L}_{\{\cup,\cap,\Rightarrow,\neg\}}$ by Greek capital letters

$$\Gamma, \Delta, \Sigma, \quad \ldots,$$

with indices if necessary. We define a sequent as follows.

def:seq **Definition 6 (Sequent)**

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

$$\Gamma \longrightarrow \Delta$$

is called a **sequent**. Γ *is called the* **antecedent** *of the sequent,* Δ *is called the* **succedent**, *and each formula in* Γ *and* Δ *is called a* **sequent-formula**.

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 Γ, Δ, we denote by σ_Γ any **conjunction** of all formulas of Γ, and by δ_Δ any **disjunction** of all formulas of Δ. The intuitive semantics for a sequent $\Gamma \longrightarrow \Delta$ (where Γ, Δ 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

Formally, let $v : VAR \longrightarrow \{T, F\}$ be a truth assignment, v^* its extension to the set of formulas $\mathcal{F}$. We extend v^* to the set

$$SQ = \{ \; \Gamma \longrightarrow \Delta : \; \Gamma, \Delta \in \mathcal{F}^* \; \} \tag{20}$$ seq

of all sequents as follows.

`seqsem` **Definition 7** *For any sequent* $\Gamma \longrightarrow \Delta \in SQ$*,*

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

In the case when $\Gamma = \emptyset$ we define: $v^*(\;\longrightarrow \Delta) \;=\; T \;\Rightarrow v^*(\delta_\Delta)$. In the case $\Delta = \emptyset$ we define $v^*(\Gamma \longrightarrow\;) \;=\; v^*(\sigma_\Gamma) \Rightarrow\; F$.

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

$$v \models \;\Gamma \;\longrightarrow\; \Delta.$$

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

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

Example 2

Let $\Gamma \longrightarrow\; \Delta$ *be a sequent*

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

Any truth assignment v*, such that* $v(a) = T$ *and* $v(b) = T$ *is a model for* $\Gamma \longrightarrow\; \Delta$*, i.e.*

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

We verify it by performing 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 $\Gamma \longrightarrow\; \Delta$. 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:

$$\frac{P_1}{C} \quad or \quad \frac{P_1 \ ; \ P_2}{C},$$

where P_1, P_2 and C are sequents. P_1, P_2 are called premisses and C 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 P by $(\circ \rightarrow)$. The notation $(\rightarrow \circ)$ means that the logical connective is introduced to the succedent of the conclusion sequent P.

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 $(\rightarrow \neg)$.

indecomp **Definition 8**

Finite sequences formed out of **positive literals** *i.e. out of propositional variables are called* **indecomposable**. *We denote them as before by*

$$\Gamma^{'}, \ \Delta^{'}, \ \ldots$$

with indices, if necessary.

A **sequent** *is* **indecomposable** *if it is formed out of indecomposable sequences, i.e. is of the form*

$$\Gamma^{'} \ \longrightarrow \ \Delta^{'}$$

for any $\Gamma^{'}, \Delta^{'} \in VAR^*$.

Remark that now the symbols $\Gamma^{'}, \ \Delta^{'}, \ \ldots$ denote sequences of **variables** (positive literals), and not sequences of literals as in (**RS**.

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

$$\Gamma'_1, \ a, \ \Gamma'_2 \ \longrightarrow \ \Delta'_1, \ a, \ \Delta'_2, \tag{21}$$ ax

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

Semantic Link

Consider axiom (21). Directly from the Definition 7 of semantics for **bf GL** we evaluate (in shorthand notation), for any truth assignments $v : VAR \longrightarrow \{T, F\}$, the following (in shorthand notation).

$$v^*(\Gamma'_1,\ a,\ \Gamma'_2\ \longrightarrow\ \Delta'_1, a, \Delta'_2) =$$

$$(\sigma_{\Gamma'_1} \cap\ a\ \cap \sigma_{\Gamma'_2})\ \Rightarrow\ (\delta_{\Delta'_1} \cup a \cup \delta_{\Delta'_2}) = T.$$

The evaluation is correct because $\models (((A \cap\ a) \cap B) \Rightarrow (C \cup a) \cup D)))$. We have thus proved the following.

`GL-axiom` **Fact 4**

Logical axioms of **GL** *are tautologies.*

Inference Rules of GL (22) `GLrules`

We adopt the following rules of inference.

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

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

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

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:

$$\mathbf{GL} = (\mathcal{L},\ SQ,\ LA,\ (\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg \Rightarrow), (\neg\neg)), \qquad (23)$$ `GL`

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 AL are the logical axioms of the system defined by the schema (21).

We define the notion of a bf 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, 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**, 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:

$$\vdash_{\mathbf{GL}} A \ \text{ if and only if } \ \vdash_{\mathbf{GL}} \ \longrightarrow \ A. \tag{24}$$ fproof

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

Proof trees

A proof tree $\Gamma \longrightarrow \Delta$ is a tree $\mathbf{T}_{\Gamma \longrightarrow \Delta}$ satisfying the following conditions:

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

2. All **leaves** 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, as we did in case of **RS** type systems, with the node on the top, and leafs on the very bottom, instead of more common way, where the leaves are on the top and root is on the bottom of the tree. We also write the proof- trees indicating additionally the name of the inference rule used at each step of the proof.

Here is a tree- proof of the de Morgan law $(\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$.

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

$$\mid (\rightarrow\Rightarrow)$$

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

$$\mid (\rightarrow \cup)$$

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

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

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

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

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

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

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

$$\bigwedge(\rightarrow \cap)$$

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

Remark 1

The proof search in **GL** *(to be defined by the decomposition tree) results are not always unique; one formula (sequent) can have many proofs.*

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

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

$$\mid (\rightarrow \Rightarrow)$$

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

$$\mid (\rightarrow \cup)$$

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

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

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

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

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

$$\bigwedge(\rightarrow \cap)$$

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

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

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

The process of searching for proofs of a formula A in **GL** consists, as in the **RS** type systems, of building decomposition trees. Their construction is similar to the one defined for **RS** type systems and is described intuitiively as follows.

We take a **root** of a decomposition tree T_A a sequent $\longrightarrow A$. For each **node**, if there is a rule of **GL** which conclusion has the same form as the node sequent, then the node has children that are premises of the rule. If the node consists only of an indecomposable sequent (built out of variables only), then it does not have any children. This is a termination condition for the decomposition tree.

We prove that each formula A generates a *finite set* $\mathcal{T}_A$ of decomposition trees, such that the following holds. If there exist a tree $T_A \in \mathcal{T}_A$ whose all leaves are axioms, then tree T_A constitutes a proof of A in **GL**. If all trees in $\mathcal{T}_A$ have at least one non-axiom leaf, the proof of A does not exist.

The first step in formally defining a notion of a decomposition tree consists of transforming the inference rules of **GL**, as we did in the case of the **RS** type systems, into corresponding *decomposition rules*.

Decomposition rules of GL

Building a proof search decomposition tree consists of using the inference rules in an inverse order; we transform the inference rules into decomposition rules by reversing the role of conclusion and its premisses. We call such rules the decomposition rules. Here are all of **GL** decomposition rules.

Conjunction decomposition rules

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

Disjunction decomposition rules

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

Implication decomposition rules

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

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

Negation decomposition rules

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

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

$(\rightarrow \cup)$ **rule**

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

$(\cup \rightarrow)$ **rule**

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

$(\rightarrow \cap)$ **rule**

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

$(\cap \rightarrow)$ **rule**

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

$(\rightarrow\Rightarrow)$ **rule**

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

$(\Rightarrow\rightarrow)$ **rule**

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

$$\bigwedge(\Rightarrow\rightarrow)$$

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

$(\neg \rightarrow)$ **rule**

$$\Gamma^{'}, \neg A, \Gamma \;\longrightarrow\; \Delta, \Delta^{'}$$

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

$$\Gamma^{'}, \Gamma \;\longrightarrow\; \Delta, A, \Delta^{'}$$

$(\longrightarrow \neg)$ **rule**

$$\Gamma^{'}, \Gamma \;\longrightarrow\; \Delta, \neg A, \Delta^{'}$$

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

$$\Gamma^{'}, A, \Gamma \;\longrightarrow\; \Delta, \Delta^{'}$$

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

We have already defined (definition 8) indecomposable sequence as any sequence $\Gamma^{'} \longrightarrow \Delta^{'}$ when $\Gamma^{'}, \Delta^{'} \in VAR^{*}$. In particular, a formula that is not a positive literal (propositional variable) is called a **decomposable formula**, and a sequent $\Gamma \longrightarrow \Delta$ where either Γ or Δ contains a decomposable formula is called a **decomposable sequent**.

By inspecting the domain of the rules we can see that at most two rules could apply for any given decomposable sequent $\Gamma \longrightarrow \Delta$.

For any decomposable sequent, at most two decomposition rules can be applied to it. This rule is determined by the first decomposable formula in Γ when we traverse it from left to right, and by the main connective of that formula, or by the first decomposable formula in Δ when we traverse it from the right to left, and by the main connective of that formula. We hence are now ready to define a decomposition tree.

Decomposition Tree $\mathbf{T}_{\rightarrow A}$

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

Step 1. The sequent $\longrightarrow A$ is the **root** of $\mathbf{T}_{\rightarrow A}$ and for any node $\Gamma \longrightarrow \Delta$ of the tree we follow the steps below.

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

Step 3. If $\Gamma \longrightarrow \Delta$ is decomposable, then we pick a decomposition rule that applies by matching the sequent of the current node with the domain of the decomposition rule. To do so we proceed as follows.

1. We traverse Γ from left to right to find the first decomposable formula. Its main connective $\circ$ identifies a possible decomposition rule $(\circ \longrightarrow)$. Then we check if this decomposition rule applies. If it does we put its conclusions (conclusion) as leaves (leaf).

2. We traverse Δ from right to left to find the first decomposable formula. Its main connective $\circ$ identifies a possible decomposition rule $(\longrightarrow\ \circ)$. Then we check if this decomposition rule applies. If it does we put its conclusions (conclusion) as leaves (leaf). 3. If 1. and 2. applies we choose one of the rules.

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

Observation 1

The decomposable $\Gamma \longrightarrow \Delta$ is always in the domain in one of the decomposition rules $(\circ \longrightarrow)$, $(\longrightarrow\ \circ)$, or in the domain of both. Hence the tree $\mathbf{T}_{\rightarrow A}$ may not be unique and all possible choices of 3. give all possible decomposition trees.

We generalize the definition of $\mathbf{T}_{\rightarrow A}$ to the decomposition tree $\mathbf{T}_{\Sigma\rightarrow\Lambda}$ of any sequent $\Sigma \longrightarrow \Lambda \in SQ$ as follows.

Decomposition Tree $\mathbf{T}_{\Sigma\longrightarrow\Lambda}$

Step 1. The sequent $\Sigma \longrightarrow \Lambda$ is the **root** of $\mathbf{T}_{\Sigma\longrightarrow\Lambda}$, and for any node $\Gamma \longrightarrow \Delta$ of the tree we follow the steps below.

Step 2. If $\Gamma \longrightarrow \Delta$ is indecomposable, then $\Gamma \longrightarrow \Delta$ becomes a **leaf** of the tree. **Step 3.** and **Step 4.** are the same as in the above definition of the tree $\mathbf{T}_{\rightarrow A}$.

Exercise 14

Prove, by constructing a proper **decomposition tree** *that*

$$\vdash_{\mathbf{GL}}((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a)).$$

Solution
By definition,we have that

$\vdash_{\mathbf{GL}}((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))$ if and only if $\vdash_{\mathbf{GL}} \longrightarrow ((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))$.

We construct a decomposition tree as follows.

$$\mathbf{T}_{\rightarrow A}$$

$$\longrightarrow ((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))$$

$$\mid (\rightarrow\Rightarrow)$$

$$(\neg a \Rightarrow b) \longrightarrow (\neg b \Rightarrow a)$$

$$\mid (\rightarrow\Rightarrow)$$

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

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

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

$$\bigwedge(\Rightarrow\longrightarrow)$$

$$\begin{array}{ccc} \longrightarrow\ \neg a, b, a & \qquad\qquad & b \longrightarrow b, a \\ \mid (\rightarrow \neg) & & axiom \\ a\ \longrightarrow\ b, a & & \\ axiom & & \end{array}$$

All leaves of the tree are axioms, hence it constitutes a **proof** in **GL**.

Exercise 15

Prove, by constructing proper **decomposition trees** *that*

$$\nvdash_{\mathbf{GL}}((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a)).$$

Solution
Observe that for some formulas A, their decomposition tree $\mathbf{T}_{\rightarrow A}$ in **GL** may not be unique. Hence we have to construct all possible decomposition trees to see that none of them is a proof, i.e. to see that each of them has a non axiom leaf. We construct the decomposition trees for $\longrightarrow A$ as follows.

$$\mathbf{T1}_{\rightarrow A}$$

$$\longrightarrow ((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))$$

(one choice)

$$| (\to\Rightarrow)$$

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

(first of two choices)

$$| (\to\Rightarrow)$$

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

(one choice)

$$| (\neg \to)$$

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

(one choice)

$$\bigwedge (\Rightarrow\longrightarrow)$$

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

$$non - axiom \qquad\qquad axiom$$

The tree contains a non- axiom leaf, hence it is **not a proof**. We have one more tree to construct.

$$\mathbf{T2}_{\to A}$$

$$\longrightarrow ((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))$$

$$| (\to\Rightarrow)$$

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

$$(second\ choice)$$

$$\bigwedge (\Rightarrow\longrightarrow)$$

$$\begin{array}{cc}
\longrightarrow\; (\neg b \Rightarrow a), a & b \longrightarrow\; (\neg b \Rightarrow a) \\
| (\longrightarrow\Rightarrow) & | (\to\Rightarrow) \\
\neg b \longrightarrow\; a, a & b, \neg b \longrightarrow\; a \\
| (\neg \to) & | (\neg \to) \\
\longrightarrow\; b, a, a & b \;\longrightarrow\; a, b \\
non - axiom & axiom
\end{array}$$

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

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

Exercise 16

Does the tree below constitute a proof in **GL***?*

$$\mathbf{T}_{\rightarrow A}$$

$$\longrightarrow \neg\neg((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))$$

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

$$\neg((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a)) \longrightarrow$$

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

$$\longrightarrow ((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))$$

$$\mid (\rightarrow\Rightarrow)$$

$$(\neg a \Rightarrow b) \longrightarrow (\neg b \Rightarrow a)$$

$$\mid (\rightarrow\Rightarrow)$$

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

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

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

$$\bigwedge(\Rightarrow\rightarrow)$$

$\longrightarrow \neg a, b, a$	$b \longrightarrow b, a$
$\mid (\rightarrow \neg)$	*axiom*
$a \longrightarrow b, a$	
axiom	

Solution
The tree above is not a proof in **GL** because a decomposition rule used in the decomposition step below **does not exists** in **GL**

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

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

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

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

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

This rule has to be replaced in by the rule:

$$(\neg \rightarrow)_1 \ \frac{\Gamma, \Gamma^{'} \ \longrightarrow \ \Delta,\ A,\ \Delta^{'}}{\Gamma,\ \neg A,\ \Gamma' \ \longrightarrow \ \Delta, \Delta'}$$

5 GL Soundness and Completeness

The system **GL** admits a constructive proof of the Completeness Theorem, similar to completeness proofs for **RS** type proof systems (Theorems 11, 6, 7). It also relays on strong soundness property of its inference rules. We are going to prove that the following holds.

ss-GL **Theorem 8 (GL Strong Soundness)**

The proof system **GL** *is strongly sound.*

Proof We have already proved (Fact 4) that logical axioms of **GL** are tautologies, so we have to prove now that its rules of inference are strongly sound (definition 4). Proofs of strong soundness of rules of inference of **GL** are more involved then the proofs for the **RS** type rules. We prove as an example the strong soundness of four of inference rules. Proofs for all other rules follows the same patterns and is left as an exercise.

By definition 4 of strong soundness we have to show the condition (15). Written formally it says that we have to show that that if $P_1,\ P_2$ are premisses of a given rule and C is its conclusion, then for all truth assignments $v : VAR \longrightarrow \{T, F\}$,

$$v^*(P_1) = v^*(C) \text{ in case of one premiss rule, and}$$

$$v^*(P_1) \cap v^*(P_2) = v^*(C), \text{in case of a two premisses rule.}$$

In order to prove it we need additional classical equivalencies listed below. You can fond a list of most basic classical equivalences in Chapter 3.

$$((A \Rightarrow B) \cap (A \Rightarrow C)) \equiv (A \Rightarrow (B \cap C))$$
$$((A \Rightarrow C) \cap (B \Rightarrow C)) \equiv ((A \cup B) \Rightarrow C)$$
$$((A \cap B) \Rightarrow C) \equiv (A \Rightarrow (\neg B \cup C))$$

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

$v^*(\Gamma^{'}, A, B, \Gamma \ \longrightarrow\ \Delta^{'}) = (v^*(\Gamma^{'}) \cap v^*(A) \cap v^*(B) \cap v^*(\Gamma)) \Rightarrow v^*(\Delta^{'}) = (v^*(\Gamma^{'}) \cap v^*(A \cap B) \cap v^*(\Gamma)) \Rightarrow v^*(\Delta^{'}) = v^*(\Gamma^{'}, (A \cap B), \Gamma \ \longrightarrow\ \Delta^{'})$

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

$v^*(\Gamma \ \longrightarrow\ \Delta, A, \Delta^{'}) \cap v^*(\Gamma \longrightarrow\ \Delta, B, \Delta^{'})$
$= (v^*(\Gamma) \Rightarrow v^*(\Delta) \cup v^*(A) \cup v^*(\Delta^{'})) \cap (v^*(\Gamma) \Rightarrow v^*(\Delta) \cup v^*(B) \cup v^*(\Delta^{'}))$

[we use : $((A \Rightarrow B) \cap (A \Rightarrow C)) \equiv (A \Rightarrow (B \cap C))$]
$= v^*(\Gamma) \Rightarrow ((v^*(\Delta) \cup v^*(A) \cup v^*(\Delta^{'})) \cap (v^*(\Delta) \cup v^*(B) \cup v^*(\Delta^{'})))$
[we use commutativity and distributivity]
$= v^*(\Gamma) \Rightarrow (v^*(\Delta) \cup (v^*(A \cap B)) \cup v^*(\Delta^{'}))$
$= v^*(\Gamma \ \longrightarrow \ \Delta, (A \cap B), \Delta^{'})$

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

$v^*(\Gamma^{'}, A, \Gamma \ \longrightarrow \ \Delta^{'}) \cap v^*(\Gamma^{'}, B, \Gamma \ \longrightarrow \Delta^{'})$
$= (v^*(\Gamma^{'}) \cap v^*(A) \cap v^*(\Gamma)) \Rightarrow v^*(\Delta^{'})) \cap (v^*(\Gamma^{'}) \cap v^*(B) \cap v^*(\Gamma)) \Rightarrow v^*(\Delta^{'}))$
[we use: $((A \Rightarrow C) \cap (B \Rightarrow C)) \equiv ((A \cup B) \Rightarrow C]$)
$= (v^*(\Gamma^{'}) \cap v^*(A) \cap v^*(\Gamma)) \cup (v^*(\Gamma^{'}) \cap v^*(B) \cap v^*(\Gamma)) \Rightarrow v^*(\Delta^{'})$
$= ((v^*(\Gamma^{'}) \cap v^*(\Gamma)) \cap v^*(A)) \cup ((v^*(\Gamma^{'}) \cap v^*(\Gamma)) \cap v^*(B)) \Rightarrow v^*(\Delta^{'})$
[we use commutativity and distributivity]
$= ((v^*(\Gamma^{'}) \cap v^*(\Gamma)) \cap (v^*(A \cup B)) \Rightarrow v^*(\Delta^{'})$
$= v^*(\Gamma^{'}, (A \cup B), \Gamma \ \longrightarrow \ \Delta^{'})$

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

$v^*(\Gamma^{'}, A, \Gamma \ \longrightarrow \ \Delta, \Delta^{'}) = v^*(\Gamma^{'}) \cap v^*(A) \cap v^*(\Gamma) \Rightarrow v^*(\Delta) \cup v^*(\Delta^{'})$
$= (v^*(\Gamma^{'}) \cap v^*(\Gamma)) \cap v^*(A) \Rightarrow v^*(\Delta) \cup v^*(\Delta^{'})$
[we use: $((A \cap B) \Rightarrow C) \equiv (A \Rightarrow (\neg B \cup C))$]
$= (v^*(\Gamma^{'}) \cap v^*(\Gamma)) \Rightarrow \neg v^*(A) \cup v^*(\Delta) \cup v^*(\Delta^{'}) = (v^*(\Gamma^{'}) \cap v^*(\Gamma)) \Rightarrow v^*(\Delta) \cup$
$v^*(\neg A) \cup v^*(\Delta^{'})$
$= v^*(\Gamma^{'}, \Gamma \ \longrightarrow \ \Delta, \neg A, \Delta^{'})$

The above shows the premises and conclusions are logically equivalent, therefore the rules of inference are strongly sound. It **ends** the proof.

Observe that the strong soundness implies soundness (not only by name!), hence we have also proved the following

GL:sound **Theorem 9 (Soundness for GL)**

For any sequent $\Gamma \longrightarrow \Delta \in SQ$,
if $\vdash_{\boldsymbol{GL}} \Gamma \longrightarrow \Delta$, *then* $\models \ \Gamma \longrightarrow \Delta$. *In particular, for any* $A \in \mathcal{F}$,
if $\vdash_{\boldsymbol{GL}} A$, *then* $\models \ A$.

We know by theorem 8 that all the rules of inference of **GL** of are strongly sound. The strong soundness of the rules means that if *at least* one of premisses of a rule is *false*, so is its conclusion. Hence given a sequent $\Gamma \longrightarrow \Delta \in SQ$, such that its decomposition tree $\mathbf{T}_{\Gamma\longrightarrow\Delta}$ has a branch ending with a non-axiom leaf. It means that any truth assignment v that make this non-axiom leaf false also falsifies all sequences on that branch, and hence falsifies the sequent $\Gamma \longrightarrow \Delta$. In particular, given a sequent $\longrightarrow A$ and its tree $\mathbf{T}_{\longrightarrow A}$, any v, such that falsifies its a non-axiom leaf is a counter-model for A. We have hence proved the following.

GL-cmodel **Theorem 10 (GL Counter Model)**

Given a sequent $\Gamma \longrightarrow \Delta$, such that its decomposition tree $\mathbf{T}_{\Gamma\longrightarrow\Delta}$ contains a non- axiom leaf L_A. Any truth assignment v that falsifies the non-axiom leaf L_A is a counter model for $\Gamma \longrightarrow \Delta$. In particular, given a formula $A \in \mathcal{F}$, and its decomposition tree $\mathbf{T}_A$ with a non-axiom leaf, this leaf let us define a counter-model for A **determined** *by the decomposition tree $\mathbf{T}_A$.*

Here is a simple exercise explaining how the construction of a counter-model determined by the decomposition tree of a works.

Exercise 17

Prove, by constructing a counter-model determined by decomposition tree that

$$\not\models \ ((b \Rightarrow a) \Rightarrow (\neg b \Rightarrow a)).$$

Solution
We construct the decomposition tree for the formula $A : ((b \Rightarrow a) \Rightarrow (\neg b \Rightarrow a))$ as follows.

$$\mathbf{T}_{\rightarrow A}$$

$$\begin{array}{c}
\longrightarrow ((b \Rightarrow a) \Rightarrow (\neg b \Rightarrow a)) \\
\mid (\rightarrow\Rightarrow) \\
(b \Rightarrow a) \ \longrightarrow \ (\neg b \Rightarrow a) \\
\mid (\rightarrow\Rightarrow) \\
\neg b, (b \Rightarrow a) \ \longrightarrow \ a \\
\mid (\neg \rightarrow) \\
(b \Rightarrow a) \ \longrightarrow \ b, a \\
\bigwedge(\Rightarrow\longrightarrow)
\end{array}$$

$$\begin{array}{cc}
\longrightarrow \ b, b, a & a \longrightarrow b, a \\
non - axiom & axiom
\end{array}$$

The non-axiom leaf L_A we want to falsify is $\longrightarrow \; b, b, a$. Let $v : VAR \longrightarrow \{T, F\}$ be a truth assignment. By definition 7 of semantic for **GL** we have that $v^*(L_A) = v^*(\longrightarrow b, b, a) = (T \Rightarrow v(b) \cup v(b) \cup v(a))$. Hence $v^*(\longrightarrow b, b, a) = F$ if and only if $(T \Rightarrow v(b) \cup v(b) \cup v(a)) = F$ if and only if $v(b) = v(a) = F$. The Theorem 10, says that the logical value **F** determined by the evaluation a non-axiom leaf L_A "climbs" the decomposition tree. We picture it as follows.

$$\mathbf{T}_{\rightarrow A}$$

$$\longrightarrow ((b \Rightarrow a) \Rightarrow (\neg b \Rightarrow a)) \quad \mathbf{F}$$

$$\mid (\rightarrow\Rightarrow)$$

$$(b \Rightarrow a) \;\longrightarrow\; (\neg b \Rightarrow a) \;\; \mathbf{F}$$

$$\mid (\rightarrow\Rightarrow)$$

$$\neg b, (b \Rightarrow a) \;\longrightarrow\; a \;\; \mathbf{F}$$

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

$$(b \Rightarrow a) \;\longrightarrow\; b, a \;\; \mathbf{F}$$

$$\bigwedge(\Rightarrow\longrightarrow)$$

$$\longrightarrow \; b, b, a \;\; \mathbf{F} \qquad\qquad\qquad a \longrightarrow b, a$$

$$non - axiom \qquad\qquad\qquad axiom$$

So, by theorem 10, any truth assignment $v : VAR \longrightarrow \{T, F\}$, such that $v(b) = v(a) = F$ falsifies the sequence $\longrightarrow A$, i.e. $v^*(\longrightarrow A) = T \Rightarrow v^*(A) = F$. This is possible only if $v^*(A) = F$. This proves that v is a counter model for A and we proved that $\not\models A$.

Our main goal is to prove the Completeness Theorem for **RS**. We prove first the Completeness Theorem for formulas $A \in \mathcal{F}$ and then we generalize it to any sequences $\Gamma \in \mathcal{F}^*$.

`thm:compl` **Theorem 11 (Completeness Theorem)**

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

$$\vdash_{\mathbf{GL}} A \quad \textit{if and only if} \quad \models \; A.$$

For any sequent $\Gamma \longrightarrow \Delta \in SQ$,

$$\vdash_{\mathbf{GL}} \Gamma \longrightarrow \Delta \quad \textit{if and only if} \quad \models \; \Gamma \longrightarrow \Delta.$$

Proof

We have already proved the Soundness Theorem 9, so we need to prove only the completeness part of it, namely to prove the implication:

$$\text{if} \quad \models A, \quad \text{then} \ \vdash_{\mathbf{GL}} A. \tag{25}$$ $\boxed{\texttt{1g}}$

We prove instead of the logically equivalent opposite implication:

$$\text{if} \quad \nvdash_{\mathbf{GL}} A \ \text{ then } \ \not\models A. \tag{26}$$ $\boxed{\texttt{2g}}$

Assume $\nvdash_{\mathbf{GL}} A$. By (24) it means that $\nvdash_{\mathbf{GL}} \longrightarrow A$. Let $\mathcal{T}_A$ be a set of all decomposition trees of $\longrightarrow A$. As $\nvdash_{\mathbf{GL}} \longrightarrow A$, each tree $\mathbf{T}_{\rightarrow A}$ in the set $\mathcal{T}_A$ has a non-axiom leaf. We choose an arbitrary $\mathbf{T}_{\rightarrow A}$ from $\mathcal{T}_A$. Let $L_A = \Gamma^{'} \longrightarrow \Delta^{'}$, be a non-axiom leaf of $\mathbf{T}_{\rightarrow A}$. We define a truth assignment $v : VAR \longrightarrow \{T, F\}$ which falsifies $\Gamma^{'} \longrightarrow \Delta^{'}$ as follows.

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

By the strong soundness of the rules of inference of **GL** and Theorem 10 it proves that $v^*(\longrightarrow A) = F$, i.e. that $\not\models \longrightarrow A$ and hence $\not\models A$.

Assume that $\Gamma \longrightarrow \Delta$ is any sequence such that $\nvdash_{\mathbf{GL}} \Gamma \longrightarrow \Delta$. But $\vdash_{\mathbf{GL}} \Gamma \longrightarrow \Delta$ if and only if $\vdash_{\mathbf{GL}} (\sigma_\Gamma \Rightarrow \delta_\Delta)$. So $\nvdash_{\mathbf{GL}} \Gamma \longrightarrow \Delta$ if and only if $\nvdash_{\mathbf{GL}} \sigma_\Gamma \Rightarrow \delta_\Delta)$. By already proven Case 1, $\not\models \sigma_\Gamma \Rightarrow \delta_\Delta)$, what is obviously equivalent to $\not\models \Gamma \longrightarrow \Delta$. This ends the proof of Case 2 and Completeness Theorem.

Gentzen Sequent Proof System G (27) $\boxed{\texttt{G}}$

The proof system **G** is in its structure the most similar to the proof system **RS2** defined by (19).

It is obtained from in the same way is a proof system obtained from **GL** by changing the indecomposable sequences $\Gamma^{'}$, $\Delta^{'}$ into any sequences $\Sigma, \ \Lambda \in \mathcal{F}^*$ in **all of the rules** of inference of **GL**.

The **logical axioms** LA remain the same; i.e. the components of **G** are as follows.

Axioms of G

As the axioms of **GL** we adopt any indecomposable 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

$$\Gamma'_1, \ a, \ \Gamma'_2 \ \longrightarrow \ \Delta'_1, \ a, \ \Delta'_2, \tag{28}$$ $\boxed{\texttt{gax}}$

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

Inference Rules of G (29) `Grules`

We adopt the following rules of inference.

Conjunction rules

$$(\cap \rightarrow)\quad \frac{\Sigma,\ A, B,\ \Gamma \ \longrightarrow\ \Lambda}{\Sigma, (A \cap B), \Gamma \ \longrightarrow\ \Lambda},$$

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

Disjunction rules

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

$$(\cup \rightarrow)\quad \frac{\Sigma, A, \Gamma \ \longrightarrow\ \Lambda\ ;\ \Sigma, B, \Gamma \ \longrightarrow\ \Lambda}{\Sigma, (A \cup B), \Gamma \ \longrightarrow\ \Lambda},$$

Implication rules

$$(\rightarrow\Rightarrow)\quad \frac{\Sigma, A, \Gamma \ \longrightarrow\ \Delta, B, \Lambda}{\Sigma, \Gamma \ \longrightarrow\ \Delta, (A \Rightarrow B), \Lambda},$$

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

Negation rules

$$(\neg \rightarrow)\quad \frac{\Sigma, \Gamma \ \longrightarrow\ \Delta, A, \Lambda}{\Sigma, \neg A, \Gamma \ \longrightarrow\ \Delta, \Lambda}, \qquad (\rightarrow \neg)\quad \frac{\Sigma, A, \Gamma \ \longrightarrow\ \Delta, \Lambda}{\Sigma, \Gamma \ \longrightarrow\ \Delta, \neg A, \Lambda},$$

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

Exercise 18 *Follow the example of the* **GL** *system and adopt all needed definitions and proofs to prove the completeness of the system* **G**.

Solution
We leave it to the reader to fill in details .In particular, one has to accomplish the steps below.

1. Explain why the system **G** is strongly sound. You can use the strong soundness of the system **GL** .

2. Prove, as an example, a strong soundness of 4 rules of **G**.

3. Prove the the following Strong Soundness Theorem for **G**.

ss-G **Theorem 12**

The proof system **G** *is* **strongly sound**.

4. Define shortly, in your own words, for any formula $A \in \mathcal{F}$, its decomposition tree $\mathbf{T}_{\rightarrow A}$ in **G**.

5. Extend your definition to a decomposition tree $\mathbf{T}_{\Gamma\rightarrow\Delta}$.

6. Prove that for any $\Gamma \rightarrow \Delta \in SQ$, the decomposition tree $\mathbf{T}_{\Gamma\rightarrow\Delta}$ are finite.

7. Give an example of formulas $A, B \in \mathcal{F}$ such that that $\mathbf{T}_{\rightarrow A}$ is unigue and $\mathbf{T}_{\rightarrow B}$ is not.

8. Prove the following Counter Model Theorem for **G**.

G-cmodel **Theorem 13**

Given a sequent $\Gamma \longrightarrow \Delta$, such that its decomposition tree $\mathbf{T}_{\Gamma\longrightarrow\Delta}$ contains a non- axiom leaf L_A. Any truth assignment v that falsifies the non-axiom leaf L_A is a counter model for $\Gamma \longrightarrow \Delta$.

10. Prove the following Completeness Theorem for **G**.

G-compl **Theorem 14** *For any formula $A \in \mathcal{F}$,*
1. $\vdash_{\mathbf{G}} A$ *if and only if* $\models A$,
and for any sequent $\Gamma \longrightarrow \Delta \in SQ$,
2. $\vdash_{\mathbf{G}} \Gamma \longrightarrow \Delta$ *if and only if* $\models \Gamma \longrightarrow \Delta$.

6 Original Gentzen Systems LK, LI Completeness and Hauptzatz Theorems

sec:LK

The original systems **LK** and **LI** were created by Gentzen in 1935 for classical and intuitionistic predicate logics, respectively. The proof system **LI** for intuitionistic propositional logic is a particular case of the proof system **LK**.

Both systems **LK** and **LI** have two groups of inference rules and a special rule called a *cut rule*. One group consists of a set of rules similar to the rules of systems **GL** and **G**. We call them Logical Rules. The other group contains a new type of rules, called Structural Rules. The *cut* rule in Gentzen sequent systems corresponds to the Modus Ponens rule in Hilbert proof systems as Modus Ponens is a particular case of the cut rule. The cut rule is needed to carry the original Gentzen proof of the completeness of the system **LK** and proving the adequacy of **LI** system for intituitionistic logic. Gentzen proof of completeness of **LK** was not direct. He used the completeness of already known Hilbert proof systems H and proved that any formula provable in the systems H is also provable in **LK**, respectively. Hence the need of the cut rule.

For the system **LI** he proved only the adequacy of **LI** system for intituitionistic logic since the semantics for the intuitionistic logic didn't yet exist. He used the acceptance of the Heying intuitionistic axiom system as the definition of the intuitionistic logic and proved that any formula provable in the Heyting system is also provable in **LI**.

Observe that by presence of the cut rule, Gentzen **LK, LI** systems are also a Hilbert system. What distinguishes it from all other known Hilbert proof systems is the fact that the cut rule could be eliminated from it.

This is Gentzen famous *Hauptzatz Theorem*, also called *Cut Elimination Theorem.* The elimination of the cut rule and the structure of other rules makes it possible to define effective automatic procedures for proof search, what is impossible in a case of the Hilbert style systems.

Gentzen, in his proof of Hauptzatz Theorem, developed a powerful technique adaptable to other logics. We present it here in classical propositional case and show how to adapt it to the intuitionistic case.

Gentzen proof is purely syntactical. It defines a constructive method of transformation of any formal proof (derivation) of a sequent $\Gamma \longrightarrow \Delta$ that uses a cut rule (and other rules) into its proof without use of the cut rule. Hence the English name *Cut Elimination Theorem.*

The completeness (with respect to algebraic semantics defined in chapter **??**) of the cut free system **LI** follows directly from LI Hauptzatz Theorem 22 and the intuitionistic completeness theorem (chapter **??**). The proof is a straightforward adaptation of the proof of cut free **LK** Completeness Theorem 23 and is left as a homework exercise in chapter **??**.

Rasiowa and Sikorski method of proving completeness theorem by constructing counter-models on the decomposition trees is a semantical equivalence to purely syntactical Gentzen proof of cut elimination. It is relatively simple, powerful and easy to understand. It was the reason it was first to be presented here. But it is more difficult and sometimes impossible to apply (generalize) to many non-classical logics then Gentzen cut elimination method. Moreover the Gentzen method is more flexible and in this sense more general and powerful. This is why we preset it here.

Components of **LK, LI** (30) `comp`

Language $\mathcal{L}$
The language is the same as the in case of **GL**, namely

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

Expressions

The set of all expressions $\mathcal{E}$ is, as before, the set

$$SQ = \{\Gamma \longrightarrow \Delta : \quad \Gamma, \Delta \in \mathcal{F}^*\} \tag{31}$$ SQ

of all sequents.

Logical Axioms
There is only one logical axiom, namely

$$A \quad \longrightarrow \quad A,$$

where A is any formula of $\mathcal{L}$.

Rules of Inference
There are two groups of rules of inference and they are defined are as follows.

GROUP ONE: STRUCTURAL RULES.

Weakening in the antecedent

$$(weak \rightarrow) \quad \frac{\Gamma \;\longrightarrow\; \Delta}{A,\; \Gamma \;\longrightarrow\; \Delta},$$

Weakening in the succedent

$$(\rightarrow weak) \quad \frac{\Gamma \;\longrightarrow\; \Delta}{\Gamma \;\longrightarrow\; \Delta,\; A},$$

Contraction in the antecedent

$$(contr \rightarrow) \quad \frac{A,\; A,\; \Gamma \;\longrightarrow\; \Delta}{A,\; \Gamma \;\longrightarrow\; \Delta},$$

Contraction in the succedent

$$(\rightarrow contr) \quad \frac{\Gamma \;\longrightarrow\; \Delta,\; A,\; A}{\Gamma \;\longrightarrow\; \Delta,\; A},$$

Exchange in the antecedent

$$(exch \rightarrow) \quad \frac{\Gamma_1,\; A,\; B,\; \Gamma_2 \;\longrightarrow\; \Delta}{\Gamma_1,\; B,\; A,\; \Gamma_2 \;\longrightarrow\; \Delta},$$

Exchange in the succedent

$$(\rightarrow exch) \quad \frac{\Delta \longrightarrow \Gamma_1,\; A,\; B,\; \Gamma_2}{\Delta \longrightarrow \Gamma_1,\; B,\; A,\; \Gamma_2},$$

Cut Rule

$$(cut) \quad \frac{\Gamma \longrightarrow \Delta,\ A \quad ; \quad A,\ \Sigma \longrightarrow \Theta}{\Gamma, \Sigma \longrightarrow \Delta, \Theta}.$$

The formula A is called a *cut formula.*

GROUP TWO: LOGICAL RULES

Conjunction

$$(\cap \rightarrow)_1 \quad \frac{A,\ \Gamma \longrightarrow \Delta}{(A \cap B),\ \Gamma \longrightarrow \Delta},$$

$$(\cap \rightarrow)_2 \quad \frac{B,\ \Gamma \longrightarrow \Delta}{(A \cap B),\ \Gamma \longrightarrow \Delta},$$

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

Disjunction

$$(\rightarrow \cup)_1 \quad \frac{\Gamma \longrightarrow \Delta,\ A}{\Gamma \longrightarrow \Delta,\ (A \cup B)},$$

$$(\rightarrow \cup)_2 \quad \frac{\Gamma \longrightarrow \Delta,\ B}{\Gamma \longrightarrow \Delta,\ (A \cup B)},$$

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

Implication

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

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

Negation

$$(\neg \rightarrow) \quad \frac{\Gamma \longrightarrow \Delta,\ A}{\neg A,\ \Gamma \longrightarrow \Delta},$$

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

def:LK **Definition 9 (Classical System LK)**

We define the classical Gentzen system **LK** *as*

$$\boldsymbol{LK} = (\mathcal{L},\ SQ,\ AL,\ \textit{Structural Rules},\ \ \textit{Cut Rule},\ \ \textit{Logical Rules}),$$

where all the components are defined by (30) above.

def:LI **Definition 10 (Intuitionistic System LI)**

We define the intuitionistic Gentzen system **LI** *as*

$$\boldsymbol{LK} = (\mathcal{L},\ ISQ,\ AL,\ \textit{I-Structural Rules},\ \ \textit{I- Cut Rule},\ \ \textit{I- Logical Rules}),$$

where ISQ is the following subset of the set SQ of all sequents (31)

$$ISQ = \{\Gamma \ \longrightarrow\ \Delta : \quad \Delta \ \textit{ consists of at most one formula }\}. \tag{32}$$ ISQ

The set ISQ is called the set of all **intuitionistic sequents**.

The I-Structural Rules, I- Cut Rule, I- Logical Rules are the **LK** *rules restricted to the set ISQ (32) of the intuitionistic sequents.*

We will study the intuitionistic system **LI** in chapter **??**. We concentrate now on then classical **LK**.

Classical System LK

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

$$\vdash_{\mathbf{LK}} A \ \text{ if and only if } \vdash_{\mathbf{LK}} \ \longrightarrow\ A. \tag{33}$$ Aproof

Proof Trees
We write formal proofs in **LK**, as we did for other Gentzen style proof systems in a form of **trees** in an "upside -down" form.

By a **proof tree** of a sequent $\Gamma \longrightarrow \Delta$ in **LK** we understand a tree

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

satisfying the following conditions:

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

2. All **leaves** 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.

The proofs are often called **derivations**. In particular, Gentzen, in his work used the term derivation we will use this notion as well. This is why we denote the proof trees by **D** (for derivation).

Finding derivations **D** in **LK** are is a more complex process, as the logical rules are different, then in **GL** and **G**. Proofs rely strongly on use of the Structural Rules. Even if we find a derivation that does not involve the Cut rule, the Structural rules are usually present. For example, a **derivation** of Excluded Middle $(A \cup \neg A)$ formula B in **LK** is as follows.

D

$$\longrightarrow (A \cup \neg A)$$

$$\mid (\to contr)$$

$$\longrightarrow (A \cup \neg A),\ (A \cup \neg A)$$

$$\mid (\to \cup)_1$$

$$\longrightarrow (A \cup \neg A),\ A$$

$$\mid (\to exch)$$

$$\longrightarrow A,\ (A \cup \neg A)$$

$$\mid (\to \cup)_1$$

$$\longrightarrow A,\ \neg A$$

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

$$A \longrightarrow A$$

$$axiom$$

Here is as yet another example a proof **P** (also cut free) of the de Morgan Law $(\neg(A \cap B) \Rightarrow (\neg A \cup \neg B))$.

P

$$\longrightarrow (\neg(A \cap B) \Rightarrow (\neg A \cup \neg B))$$

$$\mid (\to\Rightarrow)$$

$$(\neg(A \cap B) \longrightarrow (\neg A \cup \neg B))$$

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

$$\longrightarrow (\neg A \cup \neg B),\ (A \cap B)$$

$$\bigwedge(\Rightarrow\longrightarrow)$$

$\longrightarrow (\neg A \cup \neg B),\ A$	$\longrightarrow (\neg A \cup \neg B),\ B$
$\mid (\to exch)$	$\mid (\to exch)$
$\longrightarrow A, (\neg A \cup \neg B)$	$\longrightarrow B, (\neg A \cup \neg B)$
$\mid (\to \cup)_1$	$\mid (\to \cup)_1$
$\longrightarrow A, \neg A$	$\longrightarrow B, \neg B$
$\mid (\to \neg)$	$B \longrightarrow B$
$A \longrightarrow A$	$axiom$
$axiom$	

Observe that the Logical Rules are similar in their structure to the rules of the system **G** and hence admit the same proof of their soundness.

The rules $(\rightarrow \cup)_1,\ (\rightarrow \cup)_2$ and $(\rightarrow \cup)_1,\ (\rightarrow \cup)_2$ are **not strongly sound** as $A \not\equiv (A \cap B), B \not\equiv (A \cap B)$ and $A \not\equiv (A \cap B), B \not\equiv (A \cap B)$.

All other Logical Rules are **strongly sound**.

The Contraction and Exchange structural are also strongly sound as for any formulas $A, B \in \mathcal{F},\ A \equiv (A \cap A),\ A \equiv (A \cup A)$ and $(A \cap B) \equiv (B \cap A),\ (A \cap B) \equiv (B \cap A)$. The Weakening rule is sound because (we use shorthand notation) if a formula $(\Gamma \Rightarrow \Delta) = T$ then also $((A \cap \Gamma) \Rightarrow \Delta)) = T$ for any logical value of the formula A. But obviously $(\Gamma \Rightarrow \Delta) \not\equiv ((A \cap \Gamma) \Rightarrow \Delta))$, i.e. the Weakening rule is not strongly sound.

The Cut rule is sound as the fact $(\Gamma \Rightarrow (\Delta \cup A)) = T$ and $((A \cap \Sigma) \Rightarrow \Lambda) = T$ implies that $\Gamma, \Sigma \ \longrightarrow\ \Delta, \Lambda$. It is not strongly sound. Any truth assignment such that $\Gamma = T,\ \Delta = \Sigma = \Lambda = A = F$ proves that $(\Gamma\ \longrightarrow\ \Delta,\ A) \cap (A,\ \Sigma\ \longrightarrow\ \Lambda) \not\equiv (\Gamma, \Sigma\ \longrightarrow\ \Delta, \Lambda)$. Obviously, $\models A\ \ \longrightarrow\ \ A$.
We have proved that **LK** is sound and hence the following theorem holds.

LK:sound **Theorem 15 (Soundness for LK)**

For any sequent $\Gamma \longrightarrow \Delta$,

$$\textit{if } \vdash_{\boldsymbol{LK}} \Gamma \longrightarrow \Delta, \quad \textit{then} \quad \models\ \Gamma \longrightarrow \Delta.$$

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

$$\textit{if } \vdash_{\boldsymbol{LK}} A, \quad \textit{then} \quad \models\ A.$$

We follow now Gentzen way of proving completeness of **LK**. We choose any complete Hilbert proof system for the **LK** language $\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}$ and prove its equivalency with **LK**.

Gentzen referred to the Hilbert-Ackerman (1920) system (axiomatization) included in chapter **??**. We choose here the Rasiowa-Sikorski (1952) formalization R also included in chapter **??**.

We do it for two reasons. First, it reflexes a connection between classical and intuitionistic logics very much in a spirit Gentzen relationship between **LK** and **LI**.

We obtain a complete proof system I (chapter **??**) from R by just removing the last axiom A10. Second, both sets of axioms reflect the best what set of provable formulas is needed to conduct algebraic proofs of completeness of R and I, respectively.

Axioms of R (34) Raxiom

The set of logical axioms of the Hilbert style proof system RS for classical propositional logic all formulas of the forms

A1 $((A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C)))$,

A2 $(A \Rightarrow (A \cup B))$,

A3 $(B \Rightarrow (A \cup B))$,

A4 $((A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow ((A \cup B) \Rightarrow C)))$,

A5 $((A \cap B) \Rightarrow A)$,

A6 $((A \cap B) \Rightarrow B)$,

A7 $((C \Rightarrow A) \Rightarrow ((C \Rightarrow B) \Rightarrow (C \Rightarrow (A \cap B)))$,

A8 $((A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \cap B) \Rightarrow C))$,

A9 $(((A \cap B) \Rightarrow C) \Rightarrow (A \Rightarrow (B \Rightarrow C))$,

A10 $(A \cap \neg A) \Rightarrow B)$,

A11 $((A \Rightarrow (A \cap \neg A)) \Rightarrow \neg A)$,

A12 $(A \cup \neg A)$,

where $A, B, C \in \mathcal{F}$ are any formulas in $\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}$.
We adopt a Modus Ponens

$$(MP)\ \frac{A\ ;\ (A \Rightarrow B)}{B}$$

as the only inference rule.

We define **Hilbert System** R as

$$R = (\ \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}},\ \mathcal{F},\ A1 - A12,\ (MP)\ ), \tag{35}$$ Rdef

where A1 - A12 are defined by (34).

The system R is complete, i.e. we have the following.

R-compl **Theorem 16**

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

$$\vdash_R A \quad \textit{if and only if} \quad \models\ A.$$

We leave it as an exercise for the reader to show that all axioms A1 - A12 of the system R are provable in **LK**. Moreover, the Modus Ponens is a particular case of the cut rule, for Γ, Δ, Σ empty sequences and Θ containing only one element, a formula B. We call it also MP rule.

$$(MP) \quad \frac{\longrightarrow\ A \ ; \ A \longrightarrow\ B}{\longrightarrow\ B}.$$

This proves the following.

RLK **Theorem 17**

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

$$\textit{if } \vdash_R A, \textit{ then} \vdash_{\mathbf{LK}} A.$$

Directly from the above theorem 17, soundness of **LK** (theorem 15) and completeness of R (theorem 16) we get the completeness of **LK**.

LK-compl **Theorem 18 (LK Completeness)**

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

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

Here is Gentzen original formulation of the Hauptzatz Theorems, which we call also the Cut Elimination Theorem.

thm:H **Theorem 19 (Hauptzatz)** *(Classical* **LK***)*

Every derivation in **LK** *can be transformed into another* **LK** *derivation of the same sequent, in which no cuts occur.*

thm:I **Theorem 20 (Hauptzatz)** *(Intuitionistic* **LI***)*

Every derivation in **LI** *can be transformed into another* **LI** *derivation of the same sequent, in which no cuts occur.*

The proof is quite long and involved. We present here its main and most important steps. To facilitate the proof we introduce a more general form of the cut rule, called a mix rule defined as follows.

$$(mix) \quad \frac{\Gamma \longrightarrow \Delta \ \ ; \ \ \Sigma \longrightarrow \Theta}{\Gamma, \Sigma^{*} \longrightarrow \Delta^{*}, \Theta}, \qquad (36)$$ mix

where Σ^{*}, Δ^{*} are obtained from Σ, Δ by removing all occurrences of a common formula A. The formula A is now called a *mix formula*.

mixes **Example 3**

Here are some examples of an applications of the mix rule. Observe that the mix rule applies, as the cut does, to only one mix formula at the time.

$$(mix) \quad \frac{a \longrightarrow b, \neg a \quad ; \quad (b \cup c), b, b, D, b \longrightarrow}{a, (b \cup c), D \longrightarrow \neg a}$$

b is the mix formula.

$$(mix) \quad \frac{A \longrightarrow B, B, \neg A \quad ; \quad (b \cup c), B, B, D, B \longrightarrow \neg B}{A, (b \cup c), D \longrightarrow \neg A, \neg B}$$

B is the mix formula.

$$(mix) \quad \frac{A \longrightarrow B, \neg A, \neg A \quad ; \quad \neg A, B, B, \neg A, B \longrightarrow \neg B}{A, B, B \longrightarrow B, \neg B}$$

$\neg A$ is the mix formula.

Notice, that every derivation with cut may be transformed into a derivation with mix by means of a number of weakenings (multiple application of the weakening rules) and interchanges (multiple application of the exchange rules). Conversely, every mix may be transformed into a cut derivation by means of a certain number of preceding exchanges and contractions, though we do not use this fact in the proof. Observe that cut is a particular case of mix.

Proof of **Hauptzatz Theorems**
The proof for **LI** is the same as for **LK**. We must just be careful to add, at each step, the restriction to the ISQ sequences and the form of the rules. These restrictions do not alter the flow and validity of the **LK**proof. We leave it as homework exercise to the reader to re-write the proof below step by step for **LI**.

We conduct the proof in three main steps.
Step 1: we consider only derivations in which only mix rule is used.
Step 2: we consider first derivation with a certain Property H (definition 11) and prove lemma 2 for them. This lemma is the most crucial for the proof of the Hauptzatz.

`Hproperty` **Definition 11**

We say that a derivation $\mathbf{D}_{\Gamma \longrightarrow \Delta}$ of a sequent $\Gamma \longrightarrow \Delta$ has a **Property H** *if it satisfies the the following conditions.*

1. The root $\Gamma \longrightarrow \Delta$ of the derivation $\mathbf{D}_{\Gamma \longrightarrow \Delta}$ is obtained by direct use of the mix rule, i.e. the mix rule is the last rule of inference used in the proof (derivation) of $\Gamma \longrightarrow \Delta$.

2. The derivation $\mathbf{D}_{\Gamma \longrightarrow \Delta}$ does not contain any other application of the mix rule, i.e. the proof (derivation) of $\Gamma \longrightarrow \Delta$ does not contain any other application of the mix rule.

Hlemma

Lemma 2 (H lemma)

Any derivation that fulfills the **Property H** *(definition 11) may be transformed into a derivation of the same sequent) in which no mix occurs.*

Step 3: we use the H lemma 2 and to prove the the Hauptzatz as follows.

Hauptzatz proof from H lemma

Let **D** be any derivation (tree proof). Let $\Gamma \longrightarrow \Delta$ be any node on **D** such that its sub-tree $\mathbf{D}_{\Gamma \longrightarrow \Delta}$ has the PropertyH (definition 11). By H lemma 2 the sub-tree $\mathbf{D}_{\Gamma \longrightarrow \Delta}$ can be replaced by a tree $\mathbf{D^*}_{\Gamma \longrightarrow \Delta}$ in which no mix occurs. The rest of **D** remains unchanged. We repeat this procedure for each node N, such that the sub-tree $\mathbf{D}_N$ has the Property H until every application of mix rule has systematically been eliminated. This **ends** the proof of Hauptzatz provided the H lemma 2 has already been proved.

Step 2: **proof of H lemma**.
We now consider derivation tree **D** with the Property H, i.e. such that the mix rule is the last rule of inference used, and **D** does not contain any other application of the mix rule.

We define now two important notions: *degree n* and *rank r* of the derivation **D**. Observe that **D** contains only one application of mix rule, and the mix rule, contains only one *mix formula* A. Mix rule used may contain many copies of A, but there always is only one mix formula. We call is a *mix formula* of **D**.

def:degD

Definition 12

Given a derivation tree **D** *with the Property H.*
Let $A \in \mathcal{F}$ *be the mix formula of* **D**. *The degree* $n \geq 0$ *of* A *is called the* **degree** *of the derivation* **D**. *We write it as* $deg\mathbf{D} = degA = n$.

def:rank

Definition 13

Given a derivation tree **D** *with the Property H. We define the rank* r *of* **D** *as a sum of its left rank* Lr *and right rank* Rr *of* **D**, *i.e.*

$$r = Lr \ + \ Rr,$$

where:

1. the left rank Lr *of* **D** *in the largest number of consecutive nodes on the branch of* **D** *staring with the node containing the left premiss of the mix rule, such that each sequent on these nodes contains the mix formula in the* **succedent**;

2. the right rank Rr *of* **D** *in the largest number of consecutive nodes on the branch of* **D** *staring with the node containing the right premiss of the mix* *rule, such that each sequent on these nodes contains the mix formula in the* **antecedent**.

The lowest possible rank is evidently 2.
To prove the lemma we carry out *two complete inductions*, one on the *degree* n, the other on the *rank* r, of the derivation **D**.

It means we prove the lemma for a derivation of the degree n, assuming it to hold for derivations of a lower degree (in so far as there are such derivations, i.e., as long as $n \neq 0$), supposing, therefore, that derivations of lower degree can be already transformed into derivations without mix.

Furthermore, we shall begin by considering the case 1when the rank $r = 2$, and after that the case 2 when the rank $r > 2$, where we assume that the lemma already holds for derivations of the same degree, but a lower rank.

Case 1. Rank of r =2.

We present some cases and leave similar others to the reader as an exercise. Observe that first group contains cases that are especially simple in that they allow the mix to be immediately eliminated. The second group contains the most important cases since their consideration brings out the basic idea behind the whole proof, Here we use the *induction hypothesis* with respect do the *degree* of the derivation. We reduce each one of the cases to transformed derivations of a *lower degree*.

GROUP 1. Axioms and Structural Rules.

1. The left premiss of the mix rule is an axiom $A \longrightarrow A$.
Then the sub-tree of **D** containing mix is as follows.

$$A,\ \Sigma^* \ \longrightarrow\ \Delta$$

$$\bigwedge(mix)$$

$$A \longrightarrow A \qquad\qquad\qquad \Sigma \longrightarrow \Delta$$

We transform it, and replace it in **D** by

$$A,\ \Sigma^* \ \longrightarrow\ \Delta$$

$$\textit{possibly several exchanges and contractions}$$
$$\Sigma \longrightarrow \Delta$$

Such obtained $\mathbf{D}^*$ proves the same sequent and contains no mix.

2 . The right premiss of the mix rule is an axiom $A \longrightarrow A$.
This is a case **dual** to 1. We show here the dial transformation, but will leave the dual cases to the reader in the future.

Then the sub-tree of **D** containing mix is as follows.

$$\Sigma \;\longrightarrow\; \Delta^*,\; A$$
$$\bigwedge(mix)$$
$$\Sigma \longrightarrow \Delta \qquad\qquad\qquad A \longrightarrow A$$

We transform it, and replace it in **D** by

$$\Sigma \;\longrightarrow\; \Delta^*,\; A$$
$$\textit{possibly several exchanges and contractions}$$
$$\Sigma \longrightarrow \Delta$$

Such obtained $\mathbf{D}^*$ proves the same sequent and contains no mix.

Suppose that neither of premisses of mix is an axiom. As the rank r=2 , the right and left ranks are equal one. This means that in the sequents on the nodes directly below left premiss of the mix, the mix formula A does not occur in the *succedent*; in the sequents on the nodes directly below right premiss of the mix, the mix formula A does not occur in the *antecedent*.

In general, if a formula occurs in the antecedent (succedent) of a conclusion of a rule of inference, it is either obtained by a logical rule or by a contraction rule.

3. The left premiss of the mix rule is the conclusion of a contraction rule ($\to$ *contr*). The sub-tree of **D** containing mix is:

$$\Gamma,\; \Sigma^* \;\longrightarrow\; \Delta,\; \Theta$$

$$\bigwedge(mix)$$

$$\Gamma \longrightarrow \Delta,\ A \qquad\qquad \Sigma \longrightarrow \Theta$$
$$|\ (\rightarrow contr)$$
$$\Gamma \longrightarrow \Delta$$

We transform it, and replace it in **D** by

$$\Gamma,\ \Sigma^* \ \longrightarrow\ \Delta,\ \Theta$$

possibly several weakenings and exchanges

$$\Gamma \longrightarrow \Delta$$

Observe that the whole branch of **D** that starts with the node $\Sigma \longrightarrow \Theta$ disappears. Such obtained $\mathbf{D}^*$ proves the same sequent and contains no mix.

4. The right premiss of the mix rule is the conclusion of a contraction rule $(\rightarrow contr)$. It is a dual case to 3. and is left to the reader.

GROUP 2. Logical Rules.

1. The main connective of the mix formula is $\cap$, i.e. the mix formula is $(A \cap B)$. The left premiss of the mix rule is the conclusion of a rule $(\rightarrow \cap)$. The right premiss of the mix rule is the conclusion of a rule $(\cap \rightarrow)_1$.
The sub-tree **T** of **D** containing mix is:

$$\Gamma,\ \Sigma \ \longrightarrow\ \Delta,\ \Theta$$

$$\bigwedge(mix)$$

$$\Gamma \longrightarrow \Delta,\ (A \cap B) \qquad\qquad (A \cap B),\ \Sigma \longrightarrow \Theta$$
$$\bigwedge((\rightarrow \cap)) \qquad\qquad |\ (\cap \rightarrow)_1$$
$$A, \Sigma \longrightarrow \Theta$$

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

We transform **T** into $\mathbf{T}^*$ as follows.

$$\Gamma,\ \Sigma\ \longrightarrow\ \Delta,\ \Theta$$

possibly several weakenings and exchanges

$$\Gamma,\ \Sigma^*\ \longrightarrow\ \Delta^*,\ \Theta$$

$$\bigwedge(mix)$$

$$\Gamma \longrightarrow\ \Delta, A \qquad\qquad A, \Sigma \longrightarrow \Theta$$

We replace **T** by **T*** in **D** and obtain **D***. Now we can apply induction hypothesis with respect to the degree of the mix formula. The mix formula A in **D*** has a lower degree then the mix formula $(A \cap B)$ and by the inductive assumption the derivation **D***, and hence **D** may be transformed into one without mix.

2. The case when the left premiss of the mix rule is the conclusion of a rule $(\rightarrow \cap)$ and right premiss of the mix rule is the conclusion of a rule $(\cap \rightarrow)_2$

3. The main connective of the mix formula is $\cup$, i.e. the mix formula is $(A \cup B)$. The left premiss of the mix rule is the conclusion of a rule $(\rightarrow \cup)_1$ or $(\rightarrow \cup)_2$. The right premiss of the mix rule is the conclusion of a rule $(\cup \rightarrow)_1$. This is to be dealt with symmetrically to the $\cap$ cases.

4. The main connective of the mix formula is $\neg$, i.e. the mix formula is $\neg A$. The left premiss of the mix rule is the conclusion of a rule $(\rightarrow \neg)$. The right premiss of the mix rule is the conclusion of a rule $(\neg \rightarrow)$.

Here is the sub-tree **T** of **D** containing the application of the mix rule.

$$\Gamma,\ \Sigma\ \longrightarrow\ \Delta,\ \Theta$$

$$\bigwedge(mix)$$

$$\begin{array}{ccc}
\Gamma \longrightarrow\ \Delta,\ \neg A & \qquad\qquad & \neg A, \Sigma \longrightarrow\ \Theta \\
\mid (\rightarrow \neg) & & \mid (\neg \rightarrow) \\
A, \Gamma \longrightarrow\ \Delta & & \Sigma \longrightarrow\ \Theta, A
\end{array}$$

We transform **T** into **T*** as follows.

$$\Gamma,\ \Sigma\ \longrightarrow\ \Delta,\ \Theta$$

possibly several weakenings and exchanges

$$\Sigma,\ \Gamma^{*}\ \longrightarrow\ \Theta^{*},\ \Delta$$

$$\bigwedge(mix)$$

$$\Sigma \longrightarrow\ \Theta, A \qquad\qquad A, \Gamma \longrightarrow\ \Delta$$

We replace **T** by **T*** in **D** and obtain **D***. The new mix in **D*** may be eliminated by virtue of inductive assumption, and so from the derivation **D**.

5. The main connective of the mix formula is $\Rightarrow$, i.e. the mix formula is $(A \Rightarrow B)$. The left premiss of the mix rule is the conclusion of a rule $((\rightarrow\Rightarrow)$. The right premiss of the mix rule is the conclusion of a rule $(\Rightarrow\rightarrow)$.

Here is the sub-tree **T** of **D** containing the application of the mix rule.

$$\Gamma,\ \Sigma\ \longrightarrow\ \Delta,\ \Theta$$

$$\bigwedge(mix)$$

$$\Gamma \longrightarrow \Delta,\ (A \Rightarrow B) \qquad\qquad (A \Rightarrow B),\ \Sigma \longrightarrow\ \Theta$$

$$\mid (\rightarrow\Rightarrow) \qquad\qquad \bigwedge((\rightarrow \cap))$$

$$A,\ \Gamma\ \longrightarrow \Delta,\ B$$

$$\Sigma \longrightarrow \Theta,\ A \qquad B,\ \Sigma \longrightarrow \Theta,$$

We transform **T**into **T*** as follows.

$$\Gamma,\ \Sigma\ \longrightarrow\ \Delta,\ \Theta$$

possibly several weakenings and exchanges

$$\Sigma,\ \Gamma^{*}, \Sigma^{**} \longrightarrow\ \Theta^{*}, \Delta^{*}, \Theta$$

$$\bigwedge(mix)$$

$$\Sigma \longrightarrow \Theta,\ A \qquad\qquad A,\ \Gamma,\ \Sigma^{*}, \longrightarrow \Delta^{*},\ \Theta$$

$$\bigwedge(mix)$$

$$A,\ \Gamma\ \longrightarrow \Delta,\ B \quad B,\ \Sigma \longrightarrow \Theta,$$

The asteriks are, of course, intended as follows: Σ^*, Δ^* results from Σ, Δ by the omission of all formulas B; $\Gamma^*, \Sigma^{**}, \Theta^*$ results from Γ, Σ^*, Θ by the omission of all formulas A.

We replace **T** by **T*** in **D** and obtain **D***. Now we have two mixes, but both mix formulas are of a lower degree then n. We first apply the inductive to the assumption to the lower mix. Thus it can be eliminated. We can then also eliminate the upper mix. This **ends** the proof of the case of rank r=2.

Case $r > 2$.
In the case $r = 2$, we generally reduced the derivation to one of *lower degree*. Now we shall proceed to reduce the derivation to one of the *same* degree, but of a *lower rank*. This allows us to to be able to carry the induction with respect to the rank r of the derivation.

We use the inductive assuption in all cases except, as before, a case of an *axiom* or *structural rules*. In these cases the mix can be eliminated immediately, as it was eliminated in the previous case of rank $r = 2$.

In a case of *logical rules* we obtain the reduction of the mix of the lemma to derivations with mix of a lower ranks which consequently can be eleminated by the inductive assumption. We carry now proofs for two logical rules: $(\rightarrow \cap)$ and $(\cup \rightarrow$. The proof for all other rules is similar and is left to the reader. Also, we consider a case of left rank Lr= 1 and the right rank Rr = r ¿1. The symmetrical case left rank Lr = r ¿1 1 and the right rank Rr = 1 is left to the reader as an exercise.

Case: $Lr = 1, Rr = r > 1$. The right premiss of the mix is a conclusion of the rule $(\rightarrow \cap)$, i.e. it is of a form $\Gamma \longrightarrow \Delta, (A \cap B)$ and Γ contains the mix formula M. The left premiss of the mix is a sequent $\Theta \longrightarrow \Sigma$ and Θ contains the mix formula M. The end of the derivation **D**, .i.e. the sub-tree **T** of **D** containing mix is:

$$\Theta,\ \Gamma^* \longrightarrow \Sigma^*, \Delta, (A \cap B)$$

$$\bigwedge(mix)$$

$$\Theta \longrightarrow \Sigma \qquad\qquad\qquad \Gamma \longrightarrow \Delta,\ (A \cap B)$$

$$\bigwedge(\rightarrow \cap)$$

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

We transform **T** into **T*** as follows.

$$
\begin{array}{c}
\Theta,\ \Gamma^* \ \longrightarrow\ \Sigma^*, \Delta, (A \cap B) \\
\bigwedge(\rightarrow \cap) \\
\begin{array}{cc}
\begin{array}{c}
\Theta,\ \Gamma^* \ \longrightarrow\ \Sigma^*, \Delta, A \\
\bigwedge(mix) \\
\Theta \longrightarrow \Sigma \qquad \Gamma \longrightarrow\ \Delta, A
\end{array}
&
\begin{array}{c}
\Theta,\ \Gamma^* \ \longrightarrow\ \Sigma^*, \Delta, B \\
\bigwedge(mix) \\
\Theta \longrightarrow \Sigma \qquad \Gamma \longrightarrow\ \Delta, A
\end{array}
\end{array}
\end{array}
$$

We replace **T** by $\mathbf{T}^*$ in **D** and obtain $\mathbf{D}^*$. Now we have two mixes, but both have the right rank Rr = r-1 and both of them can be eliminated by the inductive assumption.

Case: $Lr = 1, Rr = r > 1$. The right premiss of the mix is a conclusion of the rule $(\cup \rightarrow$, i.e. it is of a form $(A \cup B), \Gamma \longrightarrow\ \Delta$ and Γ contains the mix formula M. The left premiss of the mix is a sequent $\Theta \longrightarrow \Sigma$ and Θ contains the mix formula M. The end of the derivation **D**, .i.e. the sub-tree **T** of **D** containing mix is:

$$
\begin{array}{c}
\Theta,\ (A \cup B)^*, \Gamma^* \ \longrightarrow\ \Sigma^*, \Delta, (A \cap B) \\
\bigwedge(mix) \\
\begin{array}{cc}
\Theta \longrightarrow \Sigma
&
\begin{array}{c}
(A \cup B)\Gamma \longrightarrow\ \Delta \\
\bigwedge(\cup \rightarrow) \\
A, \Gamma \longrightarrow\ \Delta \qquad B, \Gamma \longrightarrow\ \Delta
\end{array}
\end{array}
\end{array}
$$

$(A \cup B)^*$ stands either for or for nothing according as $(A \cup B)$ is unequal or equal to the mix formula M. The mix formula M certainly occurs in Γ. For otherwise M would been equal to $(A \cup B)$ and the right rank Rr would be equal to 1 contrary to the assumption.
We transform **T** into $\mathbf{T}^*$ as follows.

$$
\begin{array}{c}
\Theta,\ (A \cup B), \Gamma^* \ \longrightarrow\ \Sigma^*, \Delta, (A \cap B) \\
\bigwedge(\cup \rightarrow)
\end{array}
$$

$$
\begin{array}{cc}
A, \Theta,\ \Gamma^* \ \longrightarrow\ \Sigma^*, \Delta & B, \Theta,\ \Gamma^* \ \longrightarrow\ \Sigma^*, \Delta \\
\text{some weakenings, exchanges} & \text{some weakenings, exchanges} \\
\Theta, A^*, \Gamma^* \ \longrightarrow\ \Sigma^*, \Delta & \Theta, B^*, \Gamma^* \ \longrightarrow\ \Sigma^*, \Delta \\
\bigwedge(mix) & \bigwedge(mix) \\
 & \\
\Theta \longrightarrow \Sigma \qquad A, \Gamma \longrightarrow\ \Delta & \Theta \longrightarrow \Sigma \qquad B, \Gamma \longrightarrow\ \Delta
\end{array}
$$

Now we have two mixes, but both have the right rank Rr = r-1 and both of them can be eliminated by the inductive assumption. We replace **T** by $\mathbf{T}^*$ in **D** and obtain $\mathbf{D}^*$. This **ends** the proof of the Hauptzatz lemma and hence the proof of the **Hauptzatz Theorem** 19 and **Hauptzatz Theorem** 20.

Let's denote by **LK - c** and **LI - c** the systems **LK**, **LI** without the cut rule, i.e. we put

$$\mathbf{LK} - \mathbf{c} = \mathbf{LK} - \{(cut)\}. \tag{37}$$ LK-c

$$\mathbf{LI} - \mathbf{c} = \mathbf{LI} - \{(cut)\}. \tag{38}$$ LI-c

We re-write the Hauptzatz Theorems as follows.

LK-H **Theorem 21 (LK Hauptzatz)**

For every LK sequent $\Gamma \longrightarrow\ \Delta$,

$$\vdash_{LK} \Gamma \longrightarrow\ \Delta \quad \textit{if and only if} \quad \vdash_{LK-c} \Gamma \longrightarrow\ \Delta.$$

LI-H **Theorem 22 (LI Hauptzatz)**

For every LI sequent $\Gamma \longrightarrow\ \Delta$,

$$\vdash_{LK} \Gamma \longrightarrow\ \Delta \quad \textit{if and only if} \quad \vdash_{LK-c} \Gamma \longrightarrow\ \Delta.$$

This is why the cut-free Gentzen systems **LK-c** and **LI -c** are just called Gentzen **LK, LI**, respectively.

Directly from the Completeness Theorem 18 and the Hauptzatz Theorem 19 we get that the following.

LKc-compl **Theorem 23 (LK-c Completeness)**

For any sequent $\Gamma \longrightarrow \Delta$,

$$\vdash_{\mathbf{LK-c}} \Gamma \longrightarrow \Delta \quad \textit{if and only if} \quad \models\ \Gamma \longrightarrow \Delta.$$

Let **G** be the Gentzen sequents proof system defined by (27). We replace the logical axiom of **G**

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

where $a \in VAR$ is any propositional variable and $\Gamma'_1, \Gamma'_2,\ \Delta'_1,\ \Delta'_2 \in VAR^*$ are any indecomposable sequences, by a new logical axiom

$$\Gamma_1,\ A,\ \Gamma_2\ \longrightarrow\ \Delta_1,\ A,\ \Delta_2 \qquad (39)$$ GKa

for any $A \in \mathcal{F}$ and any sequences $\Gamma_1, \Gamma_2, \Delta_1, \Delta_2 \in SQ$. We call a resulting proof system **GK**, i.e. we have that

$$\mathbf{GK} = (\ \mathcal{L}_{\{\cup,\cap,\Rightarrow,\neg\}},\ SQ,\ LA,\ \mathcal{R}\ ) \qquad (40)$$ GK

where LA is the axiom (39) and $\mathcal{R}$ is the set (29) of rules of **G**.

Observe that the only difference between the systems **GK** and **G** is the form of their logical axioms, both being tautologies. Hence hence get the proof completeness of **GK** in the same way as we proved it for **G**, i.e. we have the following.

GK-compl **Theorem 24**

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

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

For any sequent $\Gamma \longrightarrow \Delta \in SQ$,

$$\vdash_{\mathbf{GK}} \Gamma \longrightarrow \Delta \quad \textit{if and only if} \quad \models \Gamma \longrightarrow \Delta.$$

By the **GK** the completeness theorem 24, **LK-c** completeness theorem 23 we get the equivalency of **GK** and the cut free **LK-c**.

Gequiv **Theorem 25 (LK, GK Equivalency)**

The proof systems **GK** *and the cut free***LK** *are equivalent, i.e for any sequent $\Gamma \longrightarrow \Delta$,*

$$\vdash_{\mathbf{LK}} \Gamma \longrightarrow \Delta \ \textit{ if and only if } \ \vdash_{\mathbf{GK}} \Gamma \longrightarrow \Delta.$$

7 Homework Problems

1. Write all proofs in **GL** of $(\neg(a \cap b) \Rightarrow (\neg a \cup \neg b)))$.
2. Find a formula which has a unique decomposition tree in **GL**.
3. Define shortly, in your own words, for any formula $A \in \mathcal{F}$, its decomposition tree $\mathbf{T}_{\rightarrow A}$ in **G**.

4. Extend your definition $\mathbf{T}_{\to A}$ in $\mathbf{G}$ to a decomposition tree $\mathbf{T}_{\Gamma\to\Delta}$.

5. Prove that for any $\Gamma \to \Delta \in SQ$, the decomposition tree $\mathbf{T}_{\Gamma\to\Delta}$ in $\mathbf{G}$ are finite.

6. Write all proofs in $\mathbf{G}$ of $(\neg(a \cap b) \Rightarrow (\neg a \cup \neg b)))$.

7. Find a formula A which has a unique decomposition tree in $\mathbf{G}$.

8. Prove strong soundness of rules $(\to \cup), (\to\Rightarrow)$ in $\mathbf{GL}$. List all logical equivalences used in the proofs.

9. Prove strong soundness of rules $(\Rightarrow\to), (\neg \to)$ in $\mathbf{GL}$. List all logical equivalences used in the proofs.

10. Prove strong soundness of rules $(\cup \to), (\to \neg), (\cap \to)$ in $\mathbf{G}$. List all logical equivalences used in the proofs.

11. Prove strong soundness of rules $(\Rightarrow\to), (\to \cup), (\Rightarrow\to)$ in $\mathbf{G}$. List all logical equivalences used in the proofs.

12. Explain why the system $\mathbf{G}$ is strongly sound.

13. Prove the following.

    For any sequent $\Gamma \longrightarrow \Delta \in SQ$,
    if $\vdash_{\mathbf{G}} \Gamma \longrightarrow \Delta$, then $\models \Gamma \longrightarrow \Delta$.

14. Given a formula $A = ((b \Rightarrow (a \cap c)) \Rightarrow (\neg(a \cup c) \Rightarrow (\neg b \cup a)))$.

    (i) Find all counter models determined by the decomposition trees of A in $\mathbf{GL}$. Explain why the definition of a counter model determined by the decomposition tree is correct.

    (ii) Find all counter models determined by the decomposition trees of A in $\mathbf{G}$. Explain why the definition of a counter model determined by the decomposition tree is correct.

15. Prove the following.

    Given a sequent $\Gamma \longrightarrow \Delta$, such that its decomposition tree $\mathbf{T}_{\Gamma\longrightarrow\Delta}$ in $\mathbf{G}$ contains a non- axiom leaf L_A. Any truth assignment v that falsifies the non-axiom leaf L_A is a counter model for $\Gamma \longrightarrow \Delta$.

16. Prove the following.

    For any sequent $\Gamma \longrightarrow \Delta \in SQ$,
    $\vdash_{\mathbf{G}} \Gamma \longrightarrow \Delta$ if and only if $\models \Gamma \longrightarrow \Delta$.

17. Let $\mathbf{LK\text{-}c} = \mathbf{LK} - \{(cut)\}$ and $\mathbf{GK}$ be proof systems defined as defined by (37) and (40), respectively

    (i) We know that $\mathbf{GK}$ is strongly sound. Prove that $\mathbf{LK\text{-}c}$ is sound but not strongly sound.

(ii) Find proofs of axioms A3, A7, and A11 of the R system (34) in **LK-c** and in **GK**, i.e. proofs of formulas $(B \Rightarrow (A \cup B))$, $((C \Rightarrow A) \Rightarrow ((C \Rightarrow B) \Rightarrow (C \Rightarrow (A \cap B)))$, and $((A \Rightarrow (A \cap \neg A)) \Rightarrow \neg A)$. Compare your results.

(iii) Find proofs of axioms A1, A8, and A9 of the R system (34) in **LK-c** and in **GK**, i.e. proofs of formulas $((A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C)))$, $((A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \cap B) \Rightarrow C))$, and $(((A \cap B) \Rightarrow C) \Rightarrow (A \Rightarrow (B \Rightarrow C))$. Compare your results.

(iv) Find proofs of axioms A1, A5, and A12 of the R system (34) in **LK-c** and in **GK**, i.e. proofs of formulas $((A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C))), ((A \cap B) \Rightarrow A)$, and $(A \cup \neg A)$. Compare your results.

18. Re- write carefully the proof of the classical Hauptzatz Theorem 19 for the case o the intuitionistic system **LI** (definition 10.

19. Define shortly, in your own words, for any formula $A \in \mathcal{F}$, its decomposition tree $\mathbf{T}_A$ in **LK-c**. Is the tree $\mathbf{T}_A$ always finite?

20. Given a formula $A = (\neg(a \cap b) \Rightarrow (\neg a \cup \neg b)))$. Construct one infinite and one infinite decomposition tree for A.

21. Describe major differences in the decomposition trees in **LK-c** and **GK**.

22. We have proved that **LK-c** and **GK** are equivalent, i.e. that for any sequent $\Gamma \longrightarrow \Delta$,

$$\vdash_{\mathbf{LK-c}} \Gamma \longrightarrow \Delta \text{ if and only if } \vdash_{\mathbf{GK}} \Gamma \longrightarrow \Delta.$$

The proof was not constructive; it was obtained from that fact that both systems are complete.

(ii) Describe a constructive procedure of transforming any proof in **GK** into a proof in **LK-c**.

(i) Transform a proof of a formula $(A \Rightarrow (A \cup B))$ in **GK** into a proof in **LK-c**.

(ii) Describe a constructive procedure of transforming any proof in **GK** into a proof in **LK-c**.

(iii) Show that the procedure of elimination of structural rules of **LK-c** leads to the rules inference of **GK** .