LOGICS FOR COMPUTER SCIENCE: Classical and Non-Classical Springer 2019

Anita Wasilewska

Chapter 6
Automated Proof Systems
Completeness of Classical Propositional Logic

CHAPTER 6 SLIDES

Chapter 6
Automated Proof Systems
Completeness of Classical Propositional Logic

Slides Set 1

PART 1: Proof System **RS**

Automated Search for Proofs: Decomposition Trees

PART 2: Proof System **RS**

Strong Soundness and Constructive Completeness

PART 3: Proof Systems **RS1, RS2**

Chapter 6
Automated Proof Systems
Completeness of Classical Propositional Logic

Slides Set 2

PART 4: Gentzen Sequent Systems **GL, G**

Strong Soundness and Constructive Completeness

Slides Set 3

PART 5: Original Gentzen Systems **LK, LI**

Classical and Intiutionistic Completeness

Hauptzatz Theorem

Chapter 6
Automated Proof Systems
Completeness of Classical Propositional Logic

Slides Set 1

PART 1: Proof System **RS**

Automated Search for Proofs: Decomposition Trees

Gentzen Style Proof Systems

Hilbert style systems are easy to **define** and admit different proofs of **Completeness Theorem**

They are difficult to use by humans, not mentioning computer

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**

Gentzen Style Proof Systems

The Gentzen type systems may be less intuitive then the Hilbert systems but they allow us to define **effective automatic** procedures for proof search, what was **impossible** in a case of the Hilbert systems

For this reason they are called **automated proof systems**

They serve as formal models of **computing** systems that **automate** the reasoning process

Gentzen Style Proof Systems

The Gentzen formalizations, as they are also called, were **invented** by **Gerald Gentzen** in 1934, hence the name

Gentzen proof systems for classical and intuitionistic **predicate logics** introduced special expressions built out of formulas and called sequents

This is why the Gentzen style systems using sequents as basic expressions are often called Gentzen sequent formalizations

Gentzen Style Proof Systems

We present in **Slides Set 2** our own Gentzen sequent systems **GL** and **G** and prove their **completeness**

We also present a propositional version of Gentzen original system **LK** and discuss the original proof of **Hauptsatz Theorem**

Hauptsatz Theorem is literally rendered as the Main Theorem and is known as **Cut-elimination Theorem**

We prove the equivalency of the **cut-free** propositional **LK** system and the **complete** proof system **G**

Gentzen Style Proof Systems

A propositional version of the historical Gentzen **original** formalization **LI** for intuitionistic logic is presented and discussed in Chapter 7

The **original** classical and intuitionistic **predicate** systems **LK** and **LI** are discussed in Chapter 9

Gentzen Style Proof Systems

The other historically **important** automated proof systems **RS** and **QRS** are due to **Rasiowa** and **Sikorski** (1960)

Rasiowa and **Sikorski** proof systems for classical propositional and predicate logic use as basic expressions **sequences** of formulas that are less complicated then the original Gentzen sequents

Rasiowa and **Sikorski** proof systems are simpler and are easier to understand then the Gentzen sequent systems

This is why the **Rasiowa** and **Sikorski** proof systems are the first to be presented here

Gentzen Style Proof Systems

Historical importance and lasting influence of **Rasiowa** and **Sikorski** work lays in the **fact** that they were the 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 **constructive** method and use it prove the **completeness** of the **RS** system and the systems **RS1** and **RS2**

Gentzen Style Proof Systems

We also generalize the constructive method developed by **Rasiowa** and **Sikorski** to the **Gentzen sequent** systems and prove the completeness of **GL** and **G**

The **completeness proof** for classical predicate logic system **RSQ** is presented in Chapter 9

RS Proof System

RS Proof System

Components of **RS**

Language

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

Expressions

We adopt as the set of expressions $\mathcal{E}$ the set $\mathcal{F}^*$ of all **finite sequences** of formulas

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

Notation

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

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

with indices if necessary.

RS Proof System

Semantic Link

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

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

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

we **denote**

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

We define as the next step a **formal semantics** for **RS**

Formal Semantics for **RS**

Formal Semantics

Let $v : VAR \longrightarrow \{T, F\}$ be a truth assignment and

v^* its classical semantics **extension** to the set of formulas $\mathcal{F}$

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

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

Formal Semantics for **RS**

Model

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

We write it as

$$v \models \Gamma$$

and call v a **model** for Γ

Counter- Model

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

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

Formal Semantics for **RS**

Tautology

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

We write

$$\models \Gamma$$

to denote that Γ is a **tautology**

Example

Example

Let Γ be a sequence

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

The truth assignment v such that

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

falsifies Γ, i.e. is a **counter-model** for Γ as shows the following computation

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

Exercise

1. 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

$$v \models \Gamma$$

2. Let Γ be a sequence

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

Prove that

$$\models \Gamma$$

Exercise

Solution

1. Γ is a sequence

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

We evaluate

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

We proved

$$v \models \Gamma$$

Exercise

Solution

2. 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 is a **tautology**

Rules of inference

Rules of inference are of the form:

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

where Γ_1, Γ_2 are called **premisses** and Γ is called the **conclusion** of the rule

Each rule of inference **introduces** a new logical connective or a negation of a logical connective

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

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

Rules of inference of **RS**

Rules of Inference

RS contains seven inference rules:

$$(\cup), \quad (\neg\cup), \quad (\cap), \quad (\neg\cap), \quad (\Rightarrow), \quad (\neg\Rightarrow), \quad (\neg\neg)$$

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

Literals

Definition

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**

Literals

We denote by

$$\Gamma^{'}, \quad \Delta^{'}, \quad \Sigma^{'} \dots$$

finite sequences (empty included) formed out of **literals** i.e

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

We will denote by

$$\Gamma, \quad \Delta, \quad \Sigma \dots$$

the elements of $\mathcal{F}^{*}$

Logical Axioms of **RS**

Logical Axioms

We adopt as an logical axiom of **RS** any sequence of **literals** which contains a propositional variable and its negation, i.e any sequence

$$\Gamma'_1,\ a,\ \Gamma'_2,\ \neg a,\ \Gamma'_3$$

$$\Gamma'_1,\ \neg a,\ \Gamma'_2,\ a,\ \Gamma'_3$$

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

We denote by LA the set of all logical axioms of **RS**

Logical Axioms of **RS**

Semantic Link

Consider axiom

$$\Gamma'_1,\ a,\ \Gamma'_2,\ \neg a,\ \Gamma'_3$$

Directly from the extension of the notion of tautology to **RS** we have that for any truth assignment $v : VAR \longrightarrow \{T, F\}$

$v^*(\Gamma'_1, \neg a, \Gamma'_2, a, \Gamma'_3) = v^*(\Gamma'_1) \cup v^*(\neg a) \cup v^*(a) \cup v^*(\Gamma'_2, \Gamma'_3) = v^*(\Gamma'_1) \cup T \cup v^*(\Gamma'_2, \Gamma'_3) = T$

The same applies to the axiom

$$\Gamma'_1,\ \neg a,\ \Gamma'_2,\ a,\ \Gamma'_3$$

We have thus proved the following

Fact

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

Inference Rules of **RS**

Disjunction rules

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

Conjunction rules

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

Inference Rules of **RS**

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

Proof System **RS**

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

$$\mathbf{RS} = (\mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}},\ \mathcal{F}^*,\ LA,\ \mathcal{R})$$

where the set of inference rules is

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

and LA is the set of logical axioms

Formal Proofs

Definition

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

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

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

$$\Gamma_1 \in LA \quad \text{and} \quad \Gamma_n = \Gamma$$

and for all $1 \leq i \leq n$

$\Gamma_i \in AL$, or Γ_i is a **conclusion** of one of the inference rules of **RS** with all its **premisses** placed in the sequence $\Gamma_1 \Gamma_2, \ \ldots, \ \Gamma_{i-1}$

Formal Proofs

When he 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

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

Formal Proofs

We write, however, the **formal proofs** in **RS** in a form of **tree proofs** rather then in a form of sequences expressions

We write a **proofs** in form of a **tree** such that

1. all leafs of the tree are **axioms**

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

3. The root is a the **therem**

Moreover, we write the **tree proofs** with the node on the **top**, and leafs on the very **bottom**

We adopt hence the following definition

Proof Trees

Definition

By a **proof tree** in **RS** of Γ we understand a tree

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

built out of $\Gamma \in \mathcal{E}$ satisfying the following conditions:

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

2. all leafs are axioms

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

Proof Trees

We picture, and write our proof trees with the **root** on the top, and the **leafs** on the very bottom

Additionally we write our **proof trees** indicating the name of the inference rule used at each step of the proof

Example

Assume that a **proof** of a sequence Γ from some three axioms was obtained by the subsequent use of the rules $(\cap), (\cup), (\cup), (\cap), (\cup)$, and $(\neg\neg), (\Rightarrow)$

We represent it as the following tree

Proof Trees

The tree $\mathbf{T}_{\Gamma}$

Γ

$\mid (\Rightarrow)$

conclusion of $(\neg\neg)$

$\mid (\neg\neg)$

conclusion of $(\cup)$

$\mid (\cup)$

conclusion of $(\cap)$

$\bigwedge (\cap)$

conclusion of $(\cap)$ | *conclusion of* $(\cup)$

$\mid (\cup)$ | $\mid (\cup)$

axiom | *conclusion of* $(\cap)$

$\bigwedge (\cap)$

axiom *axiom*

Proof Trees

The **Proof Trees** represent a certain visualization

for the proofs and proof search

Any **formal proof** in can be represented in a tree form and vice- versa

Any proof tree can be re-written in a linear form as

a previously defined **formal proof**

Example

The proof tree in **RS** of the de Morgan Law

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

is the as follows

Proof Trees

The tree $\mathbf{T}_A$

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

$$\mid (\Rightarrow)$$

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

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

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

$$\bigwedge(\cap)$$

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

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

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

Formal Proof

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

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

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

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

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

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

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

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

Example

Example

A search for the proof in **RS** of other de Morgan Law

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

consists of building a certain tree and proceeds as follows.

Example

The tree $\mathbf{T}_A$

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

$$\mid (\Rightarrow)$$

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

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

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

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

Example

We construct its formal proof , as before, written in a vertical manner

Here it is

$$a, b, \neg b$$

$$a, b, \neg a$$

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

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

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

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

Decomposition Trees

The **goal** in inventing proof systems like **RS** is to facilitates **automatic** proof search

We conduct such proof search by building what is called a **decomposition tree**

A **decomposition tree** $\mathbf{T}_A$ for the formula

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

is build as follows

Decomposition Trees

$$\mathbf{T}_A$$

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

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

$$\mid (\Rightarrow)$$

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

$$\mid (\Rightarrow)$$

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

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

$$\mid (\Rightarrow)$$

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

RS Decomposition Rules
and
Decomposition Trees

Decomposition Trees

The process of searching for a proof of a formula $A \in \mathcal{F}$ in **RS** consists of building a certain tree $\mathbf{T}_A$, called a **decomposition tree**

Building a **decomposition tree** is really a proof search

We define it by transforming the **RS** ineference **rules** into corresponding decomposition **rules**

Decomposition Rules

RS Decomposition Rules

Disjunction

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

Conjunction

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

Decomposition Rules

Implication

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

Negation

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

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

Tree Rules

We write the **Decomposition Rules** in a visual tree form as follows

Tree Rules

$(\cup)$ **rule**

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

$$\mid (\cup)$$

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

Tree Rules

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

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

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

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

$(\cap)$ **rule**

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

$$\bigwedge(\cap)$$

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

Tree Rules

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

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

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

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

$(\Rightarrow)$ **rule**

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

$$\mid (\Rightarrow)$$

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

Tree Rules

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

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

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

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

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

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

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

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

Definitions and Observations

Observe that we use the same names for the **inference** and **decomposition** rules

We do so because once the we have built a **decomposition tree** for a formula A 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 standard and useful definitions and observations.

Definitions and Observations

Definition

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

Definition

A formula A that is not a literal, i.e. $A \in \mathcal{F} - LT$ is called a **decomposable formula**

Definition

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

Definitions and Observations

Observation 1

For any **decomposable** sequence, 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

Definitions and Observations

Observation 2

If the main connective of the **first** decomposable formula

is $\cup, \cap, \Rightarrow$,

then the **decomposition rule** determined by it is

$(\cup), (\cap), (\Rightarrow)$, respectively

Observation 3

If the main connective of the **first** decomposable formula A is negation $\neg$

then the **decomposition rule** is determined by the

second connective of the formula A

The corresponding **decomposition rules** are

$(\neg\cup), (\neg\cap), (\neg\neg), (\neg \Rightarrow)$

Decomposition Lemma

Because of the importance of the **Observation 1** we re-write it in a form of the following

Decomposition Lemma

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

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

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

Decomposition Tree Definition

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

Let $A \in \mathcal{F}$, we define the **decomposition tree** $\mathbf{T}_A$ as follows

Step 1.

The formula A is the **root** of $\mathbf{T}_A$

For any other **node** Γ of the tree we follow the steps below

Step 2.

If Γ is **indecomposable** then Γ becomes a **leaf** of the tree

Decomposition Tree Definition

Step 3.

If Γ is **decomposable**, then we **traverse** Γ from **left** to **right** and identify the **first decomposable formula** B

By the **Decomposition Lemma**, there is exactly one decomposition rule determined by the main connective of B

We put its premiss as a **node below**, or its left and right premisses as the left and right **nodes below**, respectively

Step 4.

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

Decomposition Theorem

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

Decomposition Tree Theorem

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

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

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

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

Theorem

Proof

The tree $\mathbf{T}_{\Gamma}$ is unique by the **Decomposition Lemma**

It is **finite** because there is a finite number of logical connectives in Γ and all decomposition rules diminish the number of connectives

If the tree $\mathbf{T}_{\Gamma}$ has a **non- axiom** leaf it is not a proof by definition

By **1.** it also means that the proof **does not exist**

Example

Example

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

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

The formula A forms a one element **decomposable sequence**

The first decomposition rule used is determined by its main connective

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

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

Example

The first and only decomposition rule to be applied is $(\cup)$

The first segment of the decomposition tree $\mathbf{T}_A$ is

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

$$\mid (\cup)$$

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

Example

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 $\Rightarrow$ and we put the box around it

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

Example

The decomposition tree becomes now

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

$$\mid (\cup)$$

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

$$\mid (\Rightarrow)$$

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

Example

The next sequence to decompose is

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

with the first decomposable formula

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

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

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

Example

Next stage of the construction of the tree $\mathbf{T}_A$ is

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

$$\mid (\cup)$$

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

$$\mid (\Rightarrow)$$

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

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

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

Example

Finally, the complete $\mathbf{T}_A$ is

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

$$\mid (\cup)$$

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

$$\mid (\Rightarrow)$$

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

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

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

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

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

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

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

Example

All leaves of $\mathbf{T}_A$ are axioms

The tree $\mathbf{T}_A$ is a **proof** of A in **RS**, i.e.

$$\vdash_{\mathbf{RS}} \; ((a \cup b) \Rightarrow \neg a) \cup (\neg a \Rightarrow \neg c))$$

Example

Example Given a formula A and its decomposition tree $\mathbf{T}_A$

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

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

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

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

$$\mid (\Rightarrow)$$

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

Example

There is a leaf $\neg a, b, \neg a, c$ of the tree $\mathbf{T}_A$ that is **not an axiom**. By the **Decomposition Tree Theorem**

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

It means that the proof in **RS** of the formula $((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$ **does not exists**

Completeness Theorem

Our main goal is to prove the **Completeness Theorem** for **RS**

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

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

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

and then we generalize it to the following

Completeness Theorem 2 For any $\Gamma \in \mathcal{F}^*$,

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

Do do so we need to introduce a new notion of a

Strong Soundness and prove that the **RS** is strongly sound

Part 2: Strong Soundness
and
Constructive Completeness

Strong Soundness

Definition

Given a proof system

$$S = (\mathcal{L},\ \mathcal{E},\ LA,\ \mathcal{R})$$

Definition

A rule $r \in \mathcal{R}$ such that the **conjunction** of all its premisses is **logically equivalent** to its conclusion is called **strongly sound**

Definition

A proof system S is called **strongly sound** if and only if **all** its rules $r \in \mathcal{R}$ are **strongly sound**

Strong Soundness of RS

Theorem

The proof system **RS** is **strongly sound**

Proof

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

Proof for all other rules follows the same patterns and is left as an exercise

By definition of strong soundness we have to show that

If P_1, P_2 are premisses of a given rule and C is its conclusion, then for all v,

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

in case of one premiss rule and

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

in case of the two premisses rule.

Strong Soundness of RS

Consider the rule $(\cup)$

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

We evaluate:

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

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

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

Strong Soundness of RS

Consider the rule $(\neg\cup)$

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

We evaluate:

$$v^*(P_1) \cap v^*(P_2) = v^*(\Gamma', \neg A, \Delta) \cap v^*(\Gamma', \neg B, \Delta)$$

$$= (v^*(\Gamma') \cup v^*(\neg A) \cup v^*(\Delta)) \cap (v^*(\Gamma') \cup v^*(\neg B) \cup v^*(\Delta))$$

$$= (v^*(\Gamma', \Delta) \cup v^*(\neg A)) \cap (v^*(\Gamma', \Delta) \cup v^*(\neg B))$$

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

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

$$= v^*(\Gamma',\ \neg(A \cup B),\ \Delta) = v^*(C)$$

Soundness Theorem

Observe that the strong soundness notion implies soundness (not only by name!)

Obviously the LA of **RS** are **tautologies** , hence we have also proved the following

Soundness Theorem for **RS**

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

$$\text{If} \quad \vdash_{\mathbf{RS}} \Gamma, \quad \text{then} \quad \models \Gamma$$

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

$$\text{If} \quad \vdash_{\mathbf{RS}} A, \quad \text{then} \quad \models A$$

Strong Soundness

We proved that all the rules of inference of **RS** of are **strongly sound**, i.e. $C \equiv P$ and $C \equiv P_1 \cap P_2$

Strong soundness of the rules hence means that if **at least** one of premisses of a rule is **false**, so is its conclusion

Given a formula A, such that its $\mathbf{T}_A$ has a branch ending with a non-axiom **leaf**

By **strong** soundness, any v that make this non-axiom leaf **false** also falsifies all sequences on that branch and hence **falsifies** the the formula A

Counter Model Theorem

We have proved the following

Counter Model Theorem

Let $A \in \mathcal{F}$ be such that its decomposition tree $\mathbf{T}_A$ contains a **non- axiom** leaf L_A

Any truth assignment v that **falsifies** L_A is a **counter model** for A

Any truth assignment that **falsifies** a non- axiom **leaf** is called a **counter-model** for A **determined** by the decomposition tree $\mathbf{T}_A$

Counter Model Example

Consider a tree $\mathbf{T}_A$

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

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

$$\mid (\Rightarrow)$$

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

$$\mid (\Rightarrow)$$

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

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

$$\mid (\Rightarrow)$$

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

Counter Model Example

The tree $\mathbf{T}_A$ has a **non-axiom leaf**

$$L_A : \neg a, \ b, \neg a, \ c$$

We want to define a truth assignment $v : \ VAR \longrightarrow \{T, F\}$ **falsifies** this leaf L_A

Observe that v must be such that

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

$\neg v(a) \cup v(b) \cup \neg v(a) \cup v(c) = F$

It means that all components of the **disjunction** must be put to F

Counter Model Example

We hence get that v must be such that

$$v(a) = T, \quad v(b) = F, \quad v(c) = F$$

By the **Counter Model Theorem**, the v **determined** by the non-axiom leaf also **falsifies** the formula A

It proves that v is a **counter model** for A and

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

Counter Model

The **Counter Model Theorem** says that **F** determined by the non-axiom leaf "climbs" the tree $\mathbf{T}_A$

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

$$\mid (\cup)$$

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

$$\bigwedge(\cap)$$

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

$$\mid (\Rightarrow)$$

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

$$\mid (\Rightarrow)$$

$$\neg a, b, \neg a, c = \mathbf{F}$$

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

$$\mid (\Rightarrow)$$

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

axiom

Counter Model

Observe that the same counter model **construction** applies to any other non-axiom **leaf**, if exists

The other non-axiom leaf defines another **F** that also "climbs the tree" picture, and hence defines another **counter- model** for A

By **Decomposition Tree Theorem** all possible **restricted** counter-models for A are those **determined** by all non- axioms **leaves** of the $\mathbf{T}_A$

In our case the formula $\mathbf{T}_A$ has only one non-axiom leaf, and hence only one restricted **counter model**

RS Completeness Theorem

Completeness Theorem (Completeness Part)

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

$$\text{If} \quad \models A, \quad \text{then} \quad \vdash_{\mathbf{RS}} A$$

We prove instead the opposite implication

Completeness Theorem

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

Proof of **Completeness Theorem**

Assume that A is any formula is such that

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

By the **Decomposition Tree Theorem** the $\mathbf{T}_A$ contains a non-axiom leaf

The non-axiom leaf L_A **defines** a truth assignment v which **falsifies** it as follows:

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

Hence by **Counter Model Theorem** we have that v also **falsifies** A , i.e.

$$\not\models A$$

PART3:

Proof Systems **RS1** and **RS2**

RS1 Proof System

Poof System **RS1**

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

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

Expressions

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

is the set of expressions of **RS1**

Notation

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

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

with indices if necessary.

Rules of inference

RS1 contains **seven inference rules**, denoted by the same symbols as the rules of **RS**

$$(\cup), \quad (\neg\cup), \quad (\cap), \quad (\neg\cap), \quad (\Rightarrow), \quad (\neg\Rightarrow), \quad (\neg\neg)$$

The inference rules of **RS1** are quite similar to the rules of **RS**

Observe them **carefully** to see where lies the difference

Reminder

Any propositional variable, or a negation of a 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**

Literals Notation

We denote, as before, by

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

finite sequences (empty included) formed out of **literals** i.e

$$\Gamma', \Delta', \Sigma' \in LT^*$$

We will denote by

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

the elements of $\mathcal{F}^*$

Logical Axioms

We adopt all logical axioms of **RS** as the axioms of **RS1**, i.e.

$$\Gamma'_1,\ a,\ \Gamma'_2,\ \neg a,\ \Gamma'_3$$

$$\Gamma'_1,\ \neg a,\ \Gamma'_2,\ a,\ \Gamma'_3$$

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

Inference Rules of **RS1**

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

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

Inference Rules of **RS1**

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

Proof System **RS1**

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

$$\textbf{RS1} = (\mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}},\ \ \mathcal{E},\ \ LA,\ \ \mathcal{R})$$

where

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

for the inference rules is defined above and LA is the set of all logical axioms is the same as for **RS**

System **RS1**

Exercises

E1. Construct a proof in **RS1** of a formula

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

E2. Prove that **RS1** is **strongly sound**

E3. Define in your own words, for any formula A, the decomposition tree $\mathbf{T}_A$ in **RS1**

E4. Prove **Completeness Theorem** for **RS1**

E1. The decomposition tree $\mathbf{T}_A$ is a **proof** of A in **RS1** as all leaves are axioms

$$\mathbf{T}_A$$

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

$$\mid (\Rightarrow)$$

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

$$\mid (\cup)$$

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

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

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

$$\bigwedge (\cap)$$

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

Exercises Solutions

E2. Prove that **RS1** is **strongly sound**

Observe that the system **RS1** is obtained from **RS** by changing the sequence Γ' into Γ and the sequence Δ into Δ' 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 Γ' by Γ and Δ by Δ' in the the proof of **strong soundness** of each rule of **RS** to obtain the corresponding proof of **strong soundness** of corresponding rule of **RS1**

Strong Soundness of **RS1**

We do it, for example for the rule $(\cup)$ as follows

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

We evaluate:

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

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

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

Decomposition Trees in **RS1**

E3. Define in your own words, for any formula A, the decomposition tree $\mathbf{T}_A$ in **RS1**

The **definition** of the decomposition tree $\mathbf{T}_A$ is in its essence similar to the one for **RS** except for the changes which reflect the **differences** in the corresponding rules of inference

Decomposition Trees in **RS1**

Definition

To construct the decomposition tree $\mathbf{T}_A$ we follow the steps below

Step 1

Decompose formula 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 the decomposition tree construction

Completeness Theorem for **RS1**

E4. Prove the following **Completeness Theorem**

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

$$\text{If} \quad \models A, \quad \text{then} \quad \vdash_{\mathbf{RS1}} A$$

We prove instead the opposite implication

Completeness Theorem

$$\text{If} \quad \nvdash_{\mathbf{RS1}} A \quad \text{then} \quad \not\models A$$

Completeness Theorem for **RS1**

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

Fact 1: The decomposition tree $\mathbf{T}_A$ is a **proof** if and only if all leaves are axioms

Fact 2: The proof does not exist otherwise, i.e.

$\nvdash_{\mathbf{RS1}} A$ if and only if there is a non- axiom leaf on $\mathbf{T}_A$

Fact 2 holds because the tree $\mathbf{T}_A$ is unique

Proof of Completeness Theorem for **RS1**

Observe that we need **Facts 1, 2** in order to prove the **Completeness Theorem** by construction of a counter-model generated by a the a non- axiom leaf

Proof

Assume that A is any formula such that

$$\nvdash_{\mathbf{RS1}} A$$

By **Fact 2** 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:

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

This proves that

$$\not\models A$$

System **RS2** Definition

RS2 Definition

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 remind the same

Observe that now the decomposition tree may not be unique

Exercise 1

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

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

RS2 Exercises

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*

RS2 Exercises

T2$_A$

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

$$\mid (\Rightarrow)$$

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

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

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

$$\bigwedge(\cap)$$

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

$$\mid (\Rightarrow)$$

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

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

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

$$\bigwedge(\cap)$$

$$a, a, \neg a$$

axiom

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

axiom

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

$$\mid (\cup)$$

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

$$\mid (\Rightarrow)$$

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

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

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

$$\bigwedge(\cap)$$

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

axiom

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

axiom

System **RS2**

Exercise 2

Explain why the system **RS2** is **strongly sound**

You can use the soundness of the system RS

Solution

The only 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 rules** as 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**

RS2 Exercises

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 also **strongly sound**

RS2 Exercises

Exercise 3

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

RS2 Exercises

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**

RS2 Exercises

If the **node** consists only of literals (i.e. no decomposition rules to be applied),
then it **does not** have any children

The last statement defines a termination condition for the **tree**

This definition **correctly** defines a decomposition tree as it identifies and uses appropriate the **decomposition** rules

RS2 Exercises

Since in **RS2** **all** rules of inference have a sequence Γ instead of Γ' as it was defined for in **RS**, the **choice** of the decomposition rule for a node may be **not unique**

For **example** consider a **node**

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

The Γ in the **RS2** rules is a sequence of formulas, not literals, so for this **node** we can choose as a **decomposition rule** either rule $(\Rightarrow)$ or rule $(\cup)$

This leads to a non-unique **tree**

Exercise 4

Prove the **Completeness Theorem** for **RS2**

Solution

We need to prove the completeness part only, as the soundness has been already proved, i.e. we have to prove the implication: for any formula A ,

$$\text{if} \quad \nvdash_{RS2} A \quad \textit{then} \nvDash A$$

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

RS2 Exercises

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

The non-axiom leaf L_A **defines** a truth assignment v which falsifies A, as follows:

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

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, i.e.

$$\not\models A$$

RS2 Exercises

Exercise 5 Write a procedure $TREE_A$ such that for any formula A of **RS2** it produces its **unique** decomposition tree

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

}

RS2 Exercises

Exercise 6

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

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

RS system is **complete**, thus this Procedure is **complete**

Chapter 6
Automated Proof Systems
Completeness of Classical Propositional Logic

Slides Set 2

PART 4: **Gentzen Sequent** Systems **GL, G**

Strong Soundness and Constructive Completeness

Gentzen Sequent Systems **GL, G**

The book own **Gentzen** style proof systems **GL** and **G** for the classical propositional logic presented here are **inspired** by and are versions of the original (1934) **Gentzen** system **LK**

Their axioms, the rules of inference of the proof system considered here **operate** on expressions called by Gentzen, **sequents**

The **original** system **LK** is presented and discussed in detail in **Slides Set 3**

Gentzen Sequent System **GL**

The system **GL** presented here is the most similar in its **structure** to the system **RS** and is the first to be considered

GL admits a constructive proof of the **Completeness Theorem**

The proof is very similar to the proof of the **completeness** of the system **RS**

Gentzen Sequent System **GL**

GL Componenets

Language

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

$$\mathcal{L}_1 = \mathcal{L} \cup \{\longrightarrow\}$$

Gentzen Sequent System **GL**

As the next step we build expressions called **sequents**

The **sequents** are built out of finite sequences (empty included) of formulas of $\mathcal{L}_{\{\cup,\cap,\Rightarrow,\neg\}}$ and the Gentzen arrow $\longrightarrow$ as additional symbol

We **denote**, as in the **RS** type systems, the finite sequences (with indices if necessary) of of formulas of $\mathcal{L}_{\{\cup,\cap,\Rightarrow,\neg\}}$ by Greek capital letters

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

with indices if necessary

We define a **sequent** as follows

Sequent Definition

Definition

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** of the sequent

Each formula in Γ and Δ is called a **sequent formula**.

Gentzen Sequent

Intuitively, we interpret **semantically** a sequent

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

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

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

of the language $\mathcal{L}_{\{\cup,\cap,\Rightarrow,\neg\}}$

Gentzen Sequents

The sequent

$$A_1, ..., A_n \longrightarrow$$

where $m \geq 1$ means that $A_1 \cap ... \cap A_n$ yields a **contradiction**

The sequent

$$\longrightarrow B_1, ..., B_m$$

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

The empty sequent

$$\longrightarrow$$

means a **contradiction**

Gentzen Sequents

Given non empty sequences Γ, Δ

We denote by σ_Γ any conjunction of all formulas of Γ

We denote by δ_Δ any disjunction of all formulas of Δ

The **intuitive semantics** of a non- empty sequent $\Gamma \longrightarrow \Delta$ is

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

Formal semantics

Let $v : VAR \longrightarrow \{T, F\}$ be a truth assignment and v^* its extension to the set of formulas $\mathcal{F}$ of $\mathcal{L}_{\{\cup,\cap,\Rightarrow,\neg\}}$

We **extend** v^* to the set

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

of all **sequents** as follows

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

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

Formal Semantics

Special Cases

When $\Gamma = \emptyset$ or $\Delta = \emptyset$ we **define**

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

and

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

Formal Semantics

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 v is called a **model** for $\Gamma \longrightarrow \Delta$

We write

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

Formal Semantics

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 $\Gamma \longrightarrow \Delta$

We write it as

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

Formal Semantics

Tautology

A sequent $\Gamma \longrightarrow \Delta$ is a **tautology** if

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

We write it

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

Example

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

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

The truth assignment v for which

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

is a **model** for $\Gamma \longrightarrow \Delta$ as shows the following computation

$$v^*(a, (b \cap a) \longrightarrow \neg b, (b \Rightarrow a)) =$$

$$v^*(\sigma_{\{a,(b \cap a)\}}) \Rightarrow v^*(\delta_{\{\neg b,(b \Rightarrow a)\}})$$

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

$$= T \cap T \cap T \Rightarrow \neg T \cup (T \Rightarrow T) = T \Rightarrow (F \cup T) = T \Rightarrow T = T$$

Example

Observe that the truth assignment v for which

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

is the **only one** for which

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

and we proved that it is a **model** for

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

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

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

Indecomposable Sequents

Definition

Finite sequences formed out of **positive literals** i.e. out of propositional variables are called **indecomposable**

We denote them 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 $\Gamma^{'}, \Delta^{'} \in VAR^{*}$

Indecomposable Sequents

Remark

Remember that in the **GL** system the symbols

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

denote sequences of positive literals i.e. **variables**

They **do not** denote the sequences of literals as they did in the **RS** type systems

GL Components: Axioms

Logical Axioms LA

We adopt as an axiom any sequent of variables

(positive literals) which contains a propositional variable that appears

on both sides of the sequent arrow $\longrightarrow$, i.e any sequent

of the form

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

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

GL Components: Axioms

Semantic Link

Consider axiom

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

We evaluate (in shorthand notation), for any truth assignment $v : VAR \longrightarrow \{T, F\}$

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

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

The evaluation is correct because

$$\models (((A \cap\ a) \cap B) \Rightarrow (C \cup a) \cup D)))$$

We have thus proved the following.

Fact

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

GL Components: Rules

Inference rules

Let $\Gamma^{'}, \Delta^{'} \in VAR^{*}$ and $\Gamma, \Delta \in \mathcal{F}^{*}$

Conjunction rules

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

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

GL Rules

Disjunction rules

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

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

GL Rules

Implication rules

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

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

GL Rules

Negation rules

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

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

Gentzen System **GL** Definition

Definition

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

where

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

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

$$\cup \{(\neg \longrightarrow),\ (\longrightarrow \neg)\}$$

We write, as usual,

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

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

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

$$\vdash_{\mathbf{GL}} A \quad \text{if ad only if} \quad \longrightarrow A$$

Proof Trees

We consider, as we did with **RS** the proof trees for **GL**, i.e. we define

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

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

of sequents satisfying the following conditions:

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

2. All **leafs** are axioms

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

Proof Trees

Remark

The proof search in **GL** as defined by the **decomposition tree** for a given formula A **is not always unique**

We show an **example** on the next slide

Example

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Example

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Decomposition Trees

The process of **searching for proofs** of a formula A in **GL** consists, as in the **RS** type systems, of building certain trees, called **decomposition trees**

Their **construction** is similar to the one for **RS** type systems

We take a **root** of a **decomposition tree** T_A of of a formula A a sequent $\longrightarrow A$

For each **node**, if there is a rule of **GL** which conclusion has the same form as **node** sequent, then the **node** has **children** that are **premises** of the rule

If the **node** consists only of a sequent built only out of variables then it **does not** have any children

This is a termination condition for the tree

Decomposition Trees

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**

Decomposition Trees

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

Decomposition Rules of **GL**

Decomposition rules

Let $\Gamma', \Delta' \in VAR^*$ and $\Gamma, \Delta \in \mathcal{F}^*$

Conjunction rules

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

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

Decomposition Rules of **GL**

Disjunction rules

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

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

Decomposition Rules of **GL**

Implication rules

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

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

Decomposition Rules of **GL**

Negation rules

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

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

Decomposition Tree Definition

Definition

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

Step 1. The sequent $\longrightarrow A$ is the **root** of $\mathbf{T}_A$

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

Decomposition Tree Definition

Step 3. If $\Gamma \longrightarrow \Delta$ is **decomposable**

then we pick a decomposition rule that **matches** the sequent of the current node

To do so we **proceed** as follows

1. Given a node $\Gamma \longrightarrow \Delta$

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 $(\circ \longrightarrow)$ applies

If it does we put its conclusion(s) as leaf (leaves)

Decomposition Tree Definition

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 conclusion(s as leaf (leaves)

3. If **1.** and **2.** apply we **choose** one of the rules

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

Decomposition Tree Definition

Observe that a **decomposable** $\Gamma \longrightarrow \Delta$ is always in the domain of one of the **decomposition** rules $(\circ \longrightarrow)$, $(\longrightarrow \circ)$, or is in the domain of **both** of them

Hence the tree $\mathbf{T}_A$ may **not be** unique

All possible choices of **3.** give all possible **decomposition trees**

System **GL** Exercises

Exercise

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

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

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

We construct a decomposition tree $\mathbf{T}_{\rightarrow A}$ as follows

System **GL** Exercises

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

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

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

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

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

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

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

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

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

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

$$\mid (\rightarrow \neg) \qquad\qquad\qquad \textit{axiom}$$

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

axiom

All leaves of the tree are axioms, hence we have found the **proof** of A in **GL**

System **GL** Exercises

Exercise

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

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

Solution

For some formulas A, their decomposition tree $\mathbf{T}_{\rightarrow A}$ may **not be** unique

Hence we have to construct all possible **decomposition** trees to show that none of them is a **proof**, i.e. to show that each of them has a non axiom leaf.

We construct the decomposition trees for $\longrightarrow A$ as follows

System **GL** Exercises

$\mathbf{T}_{1\rightarrow A}$

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

$$\mid (\rightarrow\Rightarrow) \ (\textit{one choice})$$

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

$$\mid (\rightarrow\Rightarrow) \ (\textit{first of two choices})$$

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

$$\mid (\neg \rightarrow) \ (\textit{one choice})$$

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

$$\bigwedge (\Rightarrow\longrightarrow) \ (\textit{one choice})$$

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

$$\textit{non} - \textit{axiom} \qquad\qquad \textit{axiom}$$

The tree contains a **non- axiom** leaf, hence it is **not a proof**

We have one more tree to construct

System **GL** Exercises

$\mathbf{T}_{\mathbf{2}\to A}$

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

$$\mid (\to\Rightarrow) \ (\textit{one choice})$$

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

$$\bigwedge (\Rightarrow\longrightarrow) \ (\textit{second choice})$$

$\longrightarrow \ (\neg b \Rightarrow a), a$	$b \longrightarrow \ (\neg b \Rightarrow a)$
$\mid (\longrightarrow\Rightarrow)$ (*one choice*)	$\mid (\to\Rightarrow)$ (*one choice*)
$\neg b \longrightarrow \ a, a$	$b, \neg b \longrightarrow \ a$
$\mid (\neg \to)$ (*one choice*)	$\mid (\neg \to)$ (*one choice*)
$\longrightarrow \ b, a, a$	$b \ \longrightarrow \ b, a$
non – axiom	*axiom*

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

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

System **GL** Exercises

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Solution

The tree $\mathbf{T}_{\rightarrow A}$ is **not a proof** in **GL** because a rule corresponding to 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:

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

Exercises

Exercise 1

Write all possible proofs of

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

Exercise 2

Find a formula which has a unique decomposition tree

Exercise 3

Describe for which kind of formulas the decomposition tree is unique

GL Soundness and Completeness

GL Strong Soundness

The system **GL** admits a constructive proof of the **Completeness Theorem**, similar to completeness proofs for **RS** type proof systems

The completeness proof relays on the **strong soundness** property of the inference rules

We are going now prove the **strong soundness** property of the proof system **GL**

GL Strong Soundness

Proof of **strong soundness** property

We have already proved that logical axioms of **GL** are **tautologies**, so we have to prove now that its rules of inference are **strongly sound**

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

GL Strong Soundness

By definition of strong soundness we have to show that that for all rules of inference of **GL** the following conditions hold

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

GL Strong Soundness

We prove as an **example** the strong soundness of the following rules

$$(\cap \to), \quad (\to \cap), \quad (\cup \to), \quad (\to \neg)$$

In order to prove it we need additional classical logical **equivalencies** listed below

You can find a list of most **basic** classical equivalences in Chapter 3

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

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

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

GL Strong Soundness

Strong soundness of $(\cap \rightarrow)$

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

$$= v^*(\Gamma', A, B, \Gamma \longrightarrow \Delta')$$

$$= (v^*(\Gamma') \cap v^*(A) \cap v^*(B) \cap v^*(\Gamma)) \Rightarrow v^*(\Delta')$$

$$= (v^*(\Gamma') \cap v^*(A \cap B) \cap v^*(\Gamma)) \Rightarrow v^*(\Delta')$$

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

GL Strong Soundness

Strong soundness of $(\rightarrow \cap)$

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

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

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

[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]

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

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

GL Strong Soundness

Strong soundness of $(\cup \rightarrow)$

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

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

[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]

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

GL Strong Soundness

Strong soundness of $(\rightarrow \neg)$

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

$v^*(\Gamma', A, \Gamma \ \longrightarrow \ \Delta, \Delta')$

$= v^*(\Gamma') \cap v^*(A) \cap v^*(\Gamma) \Rightarrow v^*(\Delta) \cup v^*(\Delta')$

$= (v^*(\Gamma') \cap v^*(\Gamma)) \cap v^*(A) \Rightarrow v^*(\Delta) \cup v^*(\Delta')$

[we use: $((A \cap B) \Rightarrow C) \equiv (A \Rightarrow (\neg B \cup C))$]

$= (v^*(\Gamma') \cap v^*(\Gamma)) \Rightarrow \neg v^*(A) \cup v^*(\Delta) \cup v^*(\Delta')$

$= (v^*(\Gamma') \cap v^*(\Gamma)) \Rightarrow v^*(\Delta) \cup v^*(\neg A) \cup v^*(\Delta')$

$= v^*(\Gamma', \Gamma \ \longrightarrow \ \Delta, \neg A, \Delta')$

GL Strong Soundness

The above shows the premises and conclusions are logically **equivalent**

Therefore the four rules are **strongly sound**

This **ends** the proof

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

Soundness Theorem

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

$$\text{if } \vdash_{\mathbf{GL}} \Gamma \longrightarrow \Delta \quad \text{then]} \quad \models \Gamma \longrightarrow \Delta$$

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

$$\text{if } \vdash_{\mathbf{GL}} A \quad \text{then} \quad \models A$$

GL Strong Soundness

The strong soundness of the **rules** of inference means that if at least one of premisses of a rule is **false**, the conclusion of the rule is also **false**

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 makes this non-axiom leaf bf false also **falsifies** all sequents on that branch

Hence v **falsifies** the sequent $\Gamma \longrightarrow \Delta$

Counter Model

In particular, given a sequent

$$\longrightarrow A$$

and its **decomposition tree**

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

any v, that **falsifies** its non-axiom **leaf** is a **counter-model** for the formula A

We call such v a **counter model** determined by the **decomposition tree**

Counter Model Theorem

We have hence proved the following

Counter Model Theorem

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 $\mathbf{T}_A$

Exercise

We know that the system **GL** is **strongly sound**

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

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

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

Exercise

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

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

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

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

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

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

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

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

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

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

non – axiom   *axiom*

Exercise

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 of semantic for sequents we have that

$v^*(\longrightarrow b, b, a) = (T \Rightarrow v(b) \cup v(b) \cup v(a))$

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 **counter model** determined by the $\mathbf{T}_{\rightarrow A}$ is any

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

$$v(b) = v(a) = F$$

Counter Model Theorem

The **Counter Model Theorem**, 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

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

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

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

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

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

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

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

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

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

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

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

Counter Model Theorem

By **Counter Model Theorem**, any truth assignment

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

such that

$$v(b) = v(a) = F$$

falsifies the sequence $\longrightarrow A$

We evaluate

$$v^*(\longrightarrow A) = T \;\Rightarrow v^*(A) = F \quad \text{if and only if} \quad v^*(A) = F$$

This proves that v is a **counter model** for A and we proved that

$$\not\models A$$

GL Completeness

Our goal now is to prove the **Completeness Theorem** for **GL**

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

$$\vdash_{\mathbf{GL}} A \quad \text{if and only if} \quad \models A$$

Moreover

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

$$\vdash_{\mathbf{GL}} \Gamma \longrightarrow \Delta \quad \text{if and only if} \quad \models \Gamma \longrightarrow \Delta$$

GL Completeness

Proof

We have already proved the **Soundness Theorem**, so we only need to prove the implication:

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

We **prove** instead of the logically equivalent opposite implication:

$$\text{if } \nvdash_{\mathbf{GL}} A \text{ then } \not\models A$$

GL Completeness

Assume $\nvdash_{\mathbf{GL}} \; A$, i.e. $\nvdash_{\mathbf{GL}} \longrightarrow A$

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 $\mathbf{T}_{\rightarrow A} \in \mathcal{T}_A$

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** $L_A = \Gamma^{'} \longrightarrow \Delta^{'}$ as follows

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

By **Counter Model Theorem**

$$\not\models \; A$$

Gentzen Proof System G

Gentzen Proof system G

We obtain the proof system **G** from the system **GL** by **changing** the indecomposable sequences Γ', Δ' into any sequences $\Sigma, \Lambda \in \mathcal{F}^*$ in **all** of the rules of inference of **GL**

The **logical axioms** LA remain **the same** as in **GL**, i.e.

Axioms of G

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

where

$a \in VAR$ and $\Gamma'_1, \Gamma'_2, \Delta'_1, \Delta'_2 \in VAR^*$

Gentzen Proof System **G**

Rules of Inference

Conjunction

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

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

Disjunction

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

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

Gentzen Proof System **G**

Implication

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

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

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

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

Exercises

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

Here are steps **S1 - S10** needed to carry a full proof of the **Completeness Theorem**

We leave completion of them as series of **Exercises**

Write careful and full **solutions** for each of **S1 - S10** steps

Base them on corresponding proofs for **GL** system

System **G** Exercises

Here the steps

S1 Explain why the system **G** is strongly sound. You can use the strong soundness of the system **GL**

S2 Prove, as an example, a **strong** soundness of 4 rules of **G**

S3 Prove the the strong soundness of **G**

S4 Define shortly, in your own words, for any formula $A \in \mathcal{F}$, its **decomposition tree** $\mathbf{T}_{\to A}$

System **G** Exercises

S5 Extend your definition of $\mathbf{T}_{\rightarrow A}$ to a decomposition tree $\mathbf{T}_{\Gamma\rightarrow\Delta}$ for any $\Gamma \rightarrow \Delta \in SQ$

S6 Prove that for any $\Gamma \rightarrow \Delta \in SQ$, all decomposition trees $\mathbf{T}_{\Gamma\rightarrow\Delta}$ are **finite**

S7 Give an example of formulas $A, B \in \mathcal{F}$ such that that the tree $\mathbf{T}_{\rightarrow A}$ is **unique** and the tree $\mathbf{T}_{\rightarrow B}$ is **not unique**

System **G** Exercises

S8 Prove the following **Counter Model Theorem** for **G**

Theorem

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 $\mathbf{T}_A$

System **G** Exercises

S8 Prove the following **Completeness Theorem** for **G**

Theorem

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

$$\vdash_{\mathbf{G}} A \quad \text{if and only if} \quad \models A$$

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

$$\vdash_{\mathbf{G}} \Gamma \longrightarrow \Delta \quad \text{if and only if} \quad \models \Gamma \longrightarrow \Delta$$

Chapter 6
Automated Proof Systems
Completeness of Classical Propositional Logic

Slides Set 3

PART 5: **Original Gentzen** Systems **LK, LI**
Classical and Intiutionistic Completeness Theorem
and Hauptzatz Theorem

Original Gentzen Systems **LK, LI**

The **original** systems **LK** and **LI** were created by **Gentzen** in 1935 for classical and intuitionistic **predicate** logics, respectively

We present now their **propositional** verisons and use the same names **LK** and **LI**

The proof system **LI** for intuitionistic logic is a particular case of the proof system **LK**

Original Gentzen Systems **LK, LI**

Both systems **LK** and **LI** have **two groups** of inference **rules**

They both have a special rule called a **cut rule**

First group consists of a set of rules similar to the rules of systems **GL** and **G** callled **Logical Rules**

Second group contains a new type of rules

We call them **Structural Rules**

Original Gentzen Systems **LK, LI**

The **cut** rule in Gentzen sequent systems **corresponds** to the Modus Ponens rule in Hilbert proof systems

Modus Ponens is a particular case of the **cut** rule

The **cut** rule is needed to carry out the original Gentzen proof of the **completeness** of the system **LK** and for proving the **adequacy** of **LI** system for intituitionistic logic

Original Gentzen Systems **LK, LI**

Gentzen proof of **completeness** of **LK** was **not direct**

He used the **completeness** of already known Hilbert proof system H and proved that any formula provable in the systems H is also provable in **LK**

Hence the need of the **cut** rule

Original Gentzen Systems **LK, LI**

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 Heying intuitionistic axiom system as a **definition** of the intuitionistic logic and **proved** that any formula provable in the Heyting system is also provable in **LI**

Original Gentzen Systems **LK, LI**

Observe that by presence of the **cut** rule, Gentzen systems **LK** and **LI** are also Hilbert system

What **distinguishes** them from all other known Hilbert proof systems is the fact that the **cut rule** could be eliminated f

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 an effective automatic procedures for proof search, what is **impossible** in a case of the Hilbert style systems

Original Gentzen Systems **LK, LI**

Gentzen in his proof of **Hauptzatz Theorem** developed a powerful technique of proof **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**

The proof defines a **constructive** method of transformation of any formal **proof** (derivation) of a sequent $\Gamma \longrightarrow \Delta$ that uses the **cut** rule (and other rules) into its proof without use of the **cut** rule

Hence the English name **Cut Elimination Theorem**

Gentzen System **LK**

LK Components

Language

$$\mathcal{L} = \mathcal{L}_{\{\neg, \cap, \cup, \Rightarrow\}} \quad \text{and} \quad \mathcal{E} = SQ$$

for

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

Logical Axioms

There is only one logical axiom, namely

$$A \longrightarrow A$$

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

LK Components

Rules of Inference

Group one: **Structural Rules**

Weakening

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

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

Contraction

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

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

LK Components

Exchange

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

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

Cut Rule

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

A is called a **cut formula**

LK Components

Group Two: Logical Rules

Conjunction rules

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

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

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

LK Components

Disjunction rules

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

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

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

LK Components

Implication rules

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

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

Negation rules

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

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

LK Definition

Classical System LK

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

$$\mathbf{LK} = (\mathcal{L},\ SQ,\ LA, \mathcal{R}\ )$$

where

$$\mathcal{R} = \{\ \textbf{Structural}\ \text{Rules},\ \ \textbf{Cut}\ \text{Rule},\ \ \textbf{Logical}\ \text{Rules})$$

as defined by the **components** definitions

LI Definition

Intuitionistic System LI

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

$$\mathbf{LI} = (\mathcal{L},\ ISQ,\ AL, \mathcal{R}\ )$$

$\mathcal{R} = \{$ **I - Structural** Rules, **I - Cut** Rule, **I -Logical** Rules)

where $\mathcal{R}$ are the **LK** rules restricted to the set ISQ of the **intuitionistic sequents** defined as follows

$$ISQ = \{\Gamma \longrightarrow \Delta : \quad \Delta \quad \text{consists of at most one formula}\ \}$$

We will study the intuitionistic system **LI** in Chapter 7

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 **LK**, i.e. we write

$$\vdash_{\mathbf{LK}} A \quad \text{if and only if} \quad \vdash_{\mathbf{LK}} \longrightarrow A$$

LK Proof Trees

We write **formal proofs** in **LK**, as we did for other Gentzen style proof systems in a form of the **proof trees** defined as follows

Definition

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

Derivations in **LK**

Proofs are often called **derivations**

In particular, Gentzen, in his work used the term **derivation** for the proof and we will use this notion as well

This is why we **denote** the proof trees by **D**, for the **derivation**

Finding derivations **D** in **LK** is a complex **process**

LK **logical rules** are different, then in **GL** and **G**

Consequently, proofs rely strongly on use of the **structural rules**

Derivations in **LK**

For **example**, a **derivation** of Excluded Middle $(A \cup \neg A)$ formula in **LK** is as follows

D

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

$$\mid (\rightarrow contr)$$

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

$$\mid (\rightarrow \cup)_1$$

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

$$\mid (\rightarrow exch)$$

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

$$\mid (\rightarrow \cup)_1$$

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

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

$$A \longrightarrow A$$

axiom

Derivations in **LK**

Here is as yet another example a cut free derivation in **LK**

D

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

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

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

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

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

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

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

LK Soundness

LK Soundness

Observe that the **Logical Rules** of **LK** are similar in their structure to the rules of the system **G**

Hence **LK** **Logical Rules** admit similar proof of their **soundness**

The sound rules

$$(\rightarrow \cap)_1, \quad (\rightarrow \cap)_2 \quad \text{and} \quad (\rightarrow \cup)_1, \quad (\rightarrow \cup)_2$$

are not strongly sound because

$$A \not\equiv (A \cap B), \quad B \not\equiv (A \cap B) \quad \text{and} \quad A \not\equiv (A \cup B), \quad B \not\equiv (A \cup B)$$

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

LK Soundness

The Contraction and Exchange structural rules are **strongly sound** as for any formulas $A, B \in \mathcal{F}$,

$$A \equiv (A \cap A), \quad A \equiv (A \cup A) \quad \text{and}$$

$$(A \cap B) \equiv (B \cap A), \quad (A \cap B) \equiv (B \cap A)$$

The Weakening rule is **sound** because (we use shorthand notation)

$$\text{if} \quad (\Gamma \Rightarrow \Delta) = T \quad \text{then} \quad ((A \cap \Gamma) \Rightarrow \Delta) = T$$

for any logical value of the formula A

Obviously

$$(\Gamma \Rightarrow \Delta) \not\equiv ((A \cap \Gamma) \Rightarrow \Delta))$$

i.e. the Weakening rule **is not** strongly sound

LK Soundness

The Cut rule is **sound** as the fact that

$$(\Gamma \Rightarrow (\Delta \cup A)) = T \quad \text{and} \quad ((A \cap \Sigma) \Rightarrow \Lambda) = T$$

implies that

$$((\Gamma \cap \Sigma) \ \Rightarrow \ (\Delta \cup \Lambda)) = T$$

Cut rule **is not** strongly sound

Any truth assignment such that

$$\Gamma = T \quad \text{and} \quad \Delta = \Sigma = \Lambda = A = F$$

proves that

$$(\Gamma \ \longrightarrow \ \Delta, \ A) \cap (A, \ \Sigma \ \longrightarrow \ \Lambda) \not\equiv (\Gamma, \Sigma \ \longrightarrow \ \Delta, \Lambda)$$

LK Soundness

Obviously, the **Logical Axiom** is a tautology, i.e.

$$\models A \quad \longrightarrow \quad A$$

We have proved that **LK** is **sound** and the following theorem holds

Soundness Theorem

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

$$\text{if } \vdash_{\mathbf{LK}} \Gamma \longrightarrow \Delta, \quad \text{then } \models \; \Gamma \longrightarrow \Delta$$

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

$$\text{if } \vdash_{\mathbf{LK}} A, \quad \text{then } \models \; A$$

LK Completeness

LK Completeness

We follow Gentzen original proof of completeness of **LK**

We choose any **complete** Hilbert proof system for the **LK** language

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

and **prove**, after Gentzen, its **equivalency** with **LK**

Gentzen **referred** to the Hilbert-Ackerman (1920) system (axiomatization) included in chapter 5

We **choose** the Rasiowa-Sikorski (1952) formalization *R* also included in Chapter 5

LK Completeness

We **choose** the formalization R 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 from R by just **removing** the last axiom A12

Second, both sets of axioms reflect the best what set of rovable formulas is needed to conduct algebraic proofs of **completeness** of R and I, as discussed in Chapter 7

Hilbert System R

The set of **logical axioms** of the poof system R

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

Hilbert System R

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

where $A, B, C \in \mathcal{F}$ are any formulas

We adopt a Modus Ponens

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

as the **only** inference rule

We **define** the proof system R as

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

where A1 - A12 are **logical axioms** defined above

Hilbert System R

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

R **Completeness Theorem**

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

$$\vdash_R A \text{ if and only if } \models A$$

We leave it as an exercise to show that all axioms A1 - A12 of the system R are **provable** in **LK**

Moreover, the Modus Ponens rule of R is a **particular case** of the Cut rule, namely

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

This proves the following theorem

Hilbert System R

Provability Theorem

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

$$\text{if} \quad \vdash_R A, \quad \text{then} \quad \vdash_{\mathbf{LK}} A$$

Directly from the above provability theorem, the soundness of **LK** and the completeness of R we get the following

LK Completeness Theorem

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

$$\vdash_{\mathbf{LK}} A \quad \text{if and only if} \quad \models A$$

Hauptzatz

Hauptzatz

Here is Gentzen **original** formulation of the **Hauptzatz Theorems** for classical **LK** and intuitionistic **LI** proof systems

They are also routinely called the **Cut Elimination Theorems**

LK Hauptzatz

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

LI Hauptzatz

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

Mix Rule

Hauptzatz proof is quite long and very involved. We present its **main** and most important **steps**

To facilitate the **proof** we introduce as Gentzen did, a general form of the **cut rule**, called a **mix rule**

It is defined as follows

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

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

Mix Example

Here are some **examples** of an applications of the mix rule

Observe t hat the mix rule applies, as the cut does, to only **one** mix formula at the time

b is the mix formula in

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

B is the mix formula in

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

$\neg A$ is the mix formula in

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

Mix and Cut

Notice, that every derivation with **cut** may be transformed into a derivation with **mix**

We do so by means of a number of **weakenings** and **interchanges**, i.e. multiple application of the **weakening** rules **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 Hauptzatz proof

Observe that **cut** is a particular case of **mix**

Two Hauptzatz Theorems

There are two Hauptzatz theorems: classical **LK** **Hauptzatz** and **LI** **Hauptzatz**

The **proof** of intuitionistic **LI** **Hauptzatz** is basically **the same** as for **LK**

We must just be careful and **add,** at each step, the **restriction** made to the ISQ sequents and the form of the **LI** rules of inference. These **restrictions** do not alter the flow and validity of the **LK** proof

We discuss and present now the **proof** of **LK** **Hauptzatz**

We leave it as a homework exercise to **re-write** this proof the case of for **LI**

Proof of LK Hauptzatz

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** (to be defined) and prove an **H Lemma** for them

The **H Lemma** is the most crucial for the proof of the **Hauptzatz**

Property H

Property H

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**

It means that the **mix** rule is the last rule used in the derivation of $\Gamma \longrightarrow \Delta$

2. The derivation $\mathbf{D}_{\Gamma \longrightarrow \Delta}$ **does not** contain any other application of the **mix rule**

H Lemma

H Lemma

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

Step 3: we use the **H Lemma** and to prove the **Hauptzatz**

Proof of Hauptzatz

Step 3: **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 **Property H**

By **H Lemma** 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** has already been proved

Proof of H Lemma

Step 2: proof of **H lemma**

We consider derivation tree **D** with the **Property H**

It means that **D** is such that the **mix rule** is the last rule of inference **used** and **D does not** contain any other application of the **mix** rule

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 A. We call A the **mix formula** of **D**

We **define** two important notions: degree n and rank r of the derivation **D**

Degree of **D**

Definition

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

$$\deg \mathbf{D} = \deg A = n$$

Degree of D

Definition

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. **left rank** Lr of **D** is 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. **right rank** Rr of **D** is 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**.

Proof of H Lemma

We prove the **H Lemma** by carrying out two inductions

One on the **degree** n, the other on the **rank** r, of the derivation **D**

It means we prove the **H Lemma** for a derivation of the degree n, assuming it **to hold** for derivations of a lower degree as long as $n \neq 0$, i.e. we assume that derivations of lower degree **can** be already **transformed** into derivations without mix

Proof of H Lemma

The lowest possible **rank** is evidently 2

We begin by considering the **case 1** when the rank is $r = 2$
We carry induction with respect to the degree n of the derivation **D**

After that we examine the **case 2** when the rank is $r > 2$
and we assume that the **H Lemma** already **holds** for derivations of the same degree, but a lower rank

Proof of H Lemma

Case 1. Rank of $r = 2$

We carry induction with respect to the degree n of derivation **D**, i.e. with respect to degree $n \geq 0$ of the **mix formula**

We split the induction cases to consider in two groups

GROUP 1. Axioms and Structural Rules

GROUP 2. Logical Rules

We present now some cases of rules of inference as examples. There are some more cases presented in the chapter, and the rest are left as exercises

Proof of H Lemma

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

Proof of H Lemma

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

$$\dfrac{A \longrightarrow A \qquad\qquad \Sigma \longrightarrow \Delta}{A,\ \Sigma^{*} \longrightarrow \Delta}\ (mix)$$

Proof of H Lemma

We **transform** it, and **replace** it in the derivation tree **D** by

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

(possibly several exchanges and contractions)

$$\Sigma \longrightarrow \Delta$$

Such obtained tree **D*** proves the same sequent as **D** and contains **no mix**

Proof of H Lemma

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

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

$$\bigwedge(mix)$$

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

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

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

(possibly several exchanges and contractions)

$$\Sigma \longrightarrow \Delta$$

Such obtained **D*** proves the same sequent and contains no mix

Proof of H Lemma

Suppose that **neither** of premisses of **mix** is an axiom

As the **rank** is r=2 , the right and left **ranks** are requal 1

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

Proof of H Lemma

3. The left premiss of the **mix rule** is the conclusion of a contraction rule. The sub-tree of **D** containing **mix** is:

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

$$\bigwedge (mix)$$

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

$$\mid (\rightarrow contr)$$

$$\Gamma \longrightarrow \Delta$$

Proof of H Lemma

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

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

(possibly several weakenings and exchanges)

$$\Gamma \longrightarrow \Delta$$

Such obtained **D*** contains **no mix**

Observe that the whole **branch** of **D** that starts with the node $\Sigma \longrightarrow \Theta$ **disappears**

4. The right premiss of the **mix rule** is the conclusion of a contraction rule $(\rightarrow \mathit{contr})$. It is a dual case to **3.** s left as an exercise

Proof of H Lemma

GROUP 2. Logical Rules

1. 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:

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

$$\bigwedge (mix)$$

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

$$\bigwedge (\rightarrow \cap) \qquad\qquad \mid (\cap \rightarrow)_1$$

$$A, \Sigma \longrightarrow \Theta$$

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

Proof of H Lemma

We transform **T** into **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***

Proof of H Lemma

Now we can apply induction hypothesis with respect to the **degree** of the **mix** formula

The **mix** formula A in $\mathbf{D}^*$ has a lower degree then the **mix** formula $(A \cap B)$

By the inductive assumption the derivation $\mathbf{D}^*$, and hence the derivation $\mathbf{D}$ may be **transformed** into one without mix

2. The case when the left premiss of the**mix** rule is the conclusion of a rule $(\to \cap)$ and right premiss of the **mix** rule is the conclusion of a rule $(\cap \to)_2$ is dual to **1.** and is left as exercise

Proof of H Lemma

3. The main connective of the mix formula is $\cup$, i.e. the mix formula is $(A \cup B)$

This case is to be dealt with symmetrically to the $\cap$ cases and is presented in the book chapter 6

4. The main connective of the mix formula is $\neg$, i.e. the **mix** formula is $\neg A$

This case is also presented in the book chapter 6

We consider now a slightly more complicated case of the implication, i.e. the case of the **mix** formula $(A \Rightarrow B)$

Proof of H Lemma

5. The main connective of the **mix** formula is $\Rightarrow$, i.e. the **mix** formula is $(A \Rightarrow B)$

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

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

$$\bigwedge(mix)$$

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

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

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

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

Proof of H Lemma

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

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

(possibly several weakenings and exchanges)

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

$$\bigwedge (mix)$$

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

$$\bigwedge (mix)$$

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

Proof of H Lemma

The asteriks are, of course, intended as follows

$\Sigma^*, \ \Delta^*$ **results** from Σ, Δ by the **omission** of all formulas B

$\Gamma^*, \ \Sigma^{**}, \ \Theta^*$ **results** from $\Gamma, \ \Sigma^*, \ \Theta$ by the **omission** of all formulas A

Proof of H Lemma

We replace the sub-tree **T** by **T*** in **D** and obtain **D***

Now we have **two mixes**, but both **mix** formulas A and B are of a lower degree then n

We first apply the inductive assumption to the lower **mix** (formula B) and the lower **mix** is **eliminated**

We then apply by the inductive assumption and **eliminate** the upper **mix** (formula A)

This **ends** the proof of the **case** of the rank r=2

Proof of H Lemma

Case $r > 2$

In the case $r = 2$, we **reduced** the derivation to one of lower degree. Now we 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 $r = 2$

Proof of H Lemma

In a case of **logical rules** we obtain the reduction of the **mix** to derivations with **mix** of a lower ranks which consequently can be **eliminated** by the inductive assumption

We carry proofs for two **logical rules** $(\rightarrow \cap)$ and $(\cup \rightarrow)$

The proof for all other rules is similar and is left as exercise

We consider only the **case** of left rank Lr= 1 and right rank $Rr > 1$

The symmetrical **case** of left rank $Lr > 1$ and right rank Rr = 1 is left as an exercise

Proof of H Lemma

Case: $Lr = 1$ and $Rr = r > 1$

The right premiss of the **mix** is a conclusion of the inference rule $(\rightarrow \cap)$, i.e. it is of a form

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

where Γ contains a **mix** formula M

The left premiss of the **mix** is a sequent

$$\Theta \longrightarrow \Sigma$$

and Σ contains the **mix** formula M

Proof of H Lemma

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

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

$$\bigwedge (mix)$$

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

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

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

Proof of H Lemma

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

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

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

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

We perform **mix** on the left branch

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

$$\bigwedge(mix)$$

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

Proof of H Lemma

We perform **mix** on the right branch

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

$$\bigwedge (mix)$$

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

We replace **T** by **T*** in **D** and obtain **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

Proof of H Lemma

Case: $Lr = 1$ and $Rr = r > 1$

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 a **mix formula** M

The left premiss of the **mix** is a sequent

$$\Theta \longrightarrow \Sigma$$

and Σ contains the **mix formula** M

Proof of H Lemma

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

$$\Theta, \ (A \cup B)^{*}, \Gamma^{*} \ \longrightarrow \ \Sigma^{*}, \Delta$$

$$\bigwedge(mix)$$

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

$$\bigwedge(\cup \rightarrow)$$

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

Proof of H Lemma

$(A \cup B)^*$ stands **either** for $(A \cup B)$ **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 that $Rr > 1$

Proof of H Lemma

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

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

$$\bigwedge (\cup \rightarrow)$$

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

We perform **mix** on the left branch

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

(some weakenings, exchanges)

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

$$\bigwedge (mix)$$

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

Proof of H Lemma

We perform **mix** on the right branch

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

(some weakenings, exchanges)

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

$$\bigwedge (mix)$$

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

Proof of H Lemma

Now we have **two mixes**

But both have the right rank $Rr = r-1$ and hence both of them can be **eliminated** by the inductive assumption

We **replace** **T** by **T*** in **D** and **obtain** **D***

This **ends** the proof of the **Hauptzatz Lemma**

We have hence **completed** the proof of the **Hauptzatz Theorem**

LK and **LI** Hauptzatz Theorems

LK and **LI** Hauptzatz Theorems

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

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

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

We re-write the **Hauptzatz Theorems** as follows.

LK and **LI** Hauptzatz Theorem

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 Hauptzatz

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

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

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

LK-c Completeness

Directly from the **LK Completeness Theorem** and the **LK Hauptzatz Theorem** we get that the following.

LK-c Completeness Theorem

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

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

LK and **GK** Systems Equivalency

GK System

Let **G** be the Gentzen sequents proof system defined previously

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

$$\Gamma_1,\ A,\ \Gamma_2 \ \longrightarrow\ \Delta_1,\ A,\ \Delta_2$$

for any $A \in \mathcal{F}$ and any sequences

$$\Gamma_1, \Gamma_2, \Delta_1, \Delta_2 \in SQ$$

GK System

We call a resulting proof system **GK**, i.e. we defined it as follows

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

where LA is the new axiom defined above and $\mathcal{R}$ is the set of rules of the system **G**

Observe that the only difference between the systems**GK** and **G** is the form of their logical axioms, both being **tautologies**

We get the proof of **completeness** of **GK** in the same way as we proved it for **G**, i.e. we have the following

GK Completeness

GK Completeness Theorem

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

$$\vdash_{\mathbf{GK}} A \quad \text{if and only if} \quad \models A$$

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

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

LK and **GK** Systems Equivalency

By the **GK**, **LK-c** **Completeness Theorems** we get the **equivalency** of **GK** and the **cut free** **LK-c** proof systems

LK, GK Equivalency Theorem

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

$$\vdash_{\mathbf{LK}} \Gamma \longrightarrow \Delta \quad \text{if and only if} \quad \vdash_{\mathbf{GK}} \Gamma \longrightarrow \Delta$$