cse541
LOGIC for Computer Science

Professor Anita Wasilewska

LECTURE 7a

Chapter 7
Introduction to Intuitionistic and Modal Logics

PART 4: **Gentzen Sequent System LI**

Gentzen Sequent System **LI**

G. Gentzen formulated in 1935 a first syntactically decidable (in propositional case) proof systems for classical and intuitionistic logics

He proved their equivalence with their well established, respective Hilbert style formalizations

He **named** his classical system **LK** (K for Klassisch) and intuitionistic system **LI** (I for Intuitionistisch)

Gentzen Sequent System **LI**

In order to prove the **completeness** of the system **LK** and
to prove the **adequacy** of **LI** he introduced a special inference
rule, called **cut rule** that **corresponds** to the Modus Ponens
rule in Hilbert style proof systems

Then, as the next step he proved the now famous **Hauptzatz**,
called in English the **Cut Elimination Theorem**

Gentzen Sequent System **LI**

Gentzen original proof system **LI** is a particular case of his proof system **LK** for the classical logic

Both of them are presented in chapter 6 together with the original Gentzen's proof of the **Hauptzatz** for both, **LK** and **LI** proof systems

The elimination of the **cut** rule and the **structure** of other rules makes it possible to define effective automatic procedures for **proof** search, what is impossible in a case of the Hilbert style systems

LI Sequents

The Gentzen system **LI** is defined as follows.

Let

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

be the set of all Gentzen sequents built out of the formulas of the language

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

and the additional Gentzen arrow symbol $\longrightarrow$

We assume that all **LI** sequents are elements of a following subset *ISQ* of the set *SQ* of all sequents

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

The set *ISQ* is called the set of all **intuitionistic sequents**; the **LI** sequents

Axioms of **LI**

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

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

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

Rules of Inference of **LI**

The set inference rules of **LI** is divided into two groups : the **structural rules** and the **logical rules**

There are three **Structural Rules** of **LI**: Weakening, Contraction and Exchange

Weakening structural rule

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

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

A is called the weakening formula

Remember that Δ contains at most one formula

Rules of Inference of **LI**

Contraction structural rule

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

A is called the contraction formula

Remember that Δ contains at most one formula

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

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

Rules of Inference of **LI**

Exchange structural rule

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

Remember that Δ contains at most one formula

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

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

Rules of Inference of **LI**

Logical Rules

Conjunction rules

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

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

Remember that Δ contains at most one formula

Rules of Inference of **LI**

Disjunction rules

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

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

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

Remember that Δ contains at most one formula

Rules of Inference of **LI**

Implication rules

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

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

Remember that Δ contains at most one formula

Gentzen System LI

Negation rules

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

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

We define the Gentzen system **LI** as

$$\mathbf{LI} = (\mathcal{L},\ ISQ,\ LA,\ \text{Structural rules},\ \text{Logical rules})$$

LI Completeness

The completeness of the **cut-free** **LI** follows directly from **LI Hauptzatz** proved in chapter 6 and the **intuitionistic completeness** (Mostowski 1948)

Completeness of LI

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

$$\vdash_{LI} \Gamma \longrightarrow \Delta \quad \text{if and only of} \quad \models_{I} \Gamma \longrightarrow \Delta$$

In particular, for any formula A,

$$\vdash_{LI} A \quad \text{if and only of} \quad \models_{I} A$$

Intuitionistic Disjunction

The particular form the following theorem was stated without the proof by Gödel in 1931

The theorem proved by Gentzen in 1935 via **Hauptzatz** and we follow his proof

Intuitionistically Derivable Disjunction

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

$$\vdash_{LI} (A \cup B) \quad \text{if and only if} \quad \vdash_{LI} A \quad \text{or} \quad \vdash_{LI} B$$

In particular, a disjunction $(A \cup B)$ is intuitionistically **provable** in any proof system I if and only if either A or B is intuitionistically **provable** in I

Intuitionistic Disjunction

Proof of

$$\vdash_{LI} (A \cup B) \quad \text{if and only if} \quad \vdash_{LI} A \quad \text{or} \quad \vdash_{LI} B$$

Assume $\vdash_{LI} (A \cup B)$

This equivalent to $\vdash_{LI} \longrightarrow (A \cup B)$

The last step in the proof of $\longrightarrow (A \cup B)$ in **LI** must be the application of the rule $(\rightarrow \cup)_1$ to the sequent $\longrightarrow A$, or the application of the rule $(\rightarrow \cup)_2$ to the sequent $\longrightarrow B$

There is no other possibilities

We have proved that $\vdash_{LI} (A \cup B)$ implies $\vdash_{LI} A$ or $\vdash_{LI} B$

The inverse implication is obvious by respective applications of rules $(\rightarrow \cup)_1$ or $(\rightarrow \cup)_2$ to the sequents $\longrightarrow A$ or $\longrightarrow B$

Decomposition Trees in **LI**

Decomposition Trees in **LI**

Search for proofs in **LI** is a much more complicated process then the one in classical logic systems **RS** or **GL** defined in chapter 6

Here, as in any other Gentzen style proof system, proof search procedure consists of building the **decomposition** trees

Remark 1

In **RS** the decomposition tree $\mathbf{T}_A$ of any formula A is always unique

Remark 2

In **GL** the "blind search" defines, for any formula A a **finite** number of decomposition trees,

Nevertheless, it can be proved that the search can be reduced to examining only **one** of them, due to the absence of structural rules

Decomposition Trees in **LI**

Remark 3

In **LI** the structural rules play a **vital role** in the proof construction and hence, in the proof search

The fact that a given **decomposition** tree ends with an non-axiom leaf **does not** always imply that the proof does not exist

It might only imply that our search strategy was **not good**

The problem of **deciding** whether a given formula *A* **does**, or **does not** have a proof in **LI** becomes more complex then in the case of Gentzen system for classical logic

Decomposition Trees in **LI**

Before we define a heuristic method of **searching** for proof and **deciding** whether such a proof exists or not we make some observations

Observation 1

Logical rules of **LI** are similar to those in Gentzen type classical formalizations we already examined in previous chapters in a sense that each of them introduces a logical **connective**

Decomposition Trees in **LI**

Observation 2

The process of searching for a proof is a **decomposition** process in which we use the inverse of logical and structural rules as **decomposition** rules

For **example** the implication rule:

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

becomes an implication **decomposition** rule (we use the same name $(\rightarrow\Rightarrow)$ in both cases)

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

Decomposition Trees in **LI**

Observation 3

We write proofs as **trees**, so the proof search process is a process of building decomposition trees

To facilitate the process we write the **decomposition** rules in a tree decomposition form as follows

$$\Gamma \longrightarrow (A \Rightarrow B)$$

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

$$A, \Gamma \longrightarrow B$$

Decomposition Trees in **LI**

The two premisses rule $(\Rightarrow\longrightarrow)$ written as the tree decomposition rule becomes

$$(A \Rightarrow B), \Gamma \;\longrightarrow$$

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

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

Decomposition Trees in **LI**

The structural weakening rule written as the **decomposition** rule is

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

We write it in a **tree decomposition** form as

$$\Gamma \longrightarrow A$$

$$\mid (\rightarrow weak)$$

$$\Gamma \longrightarrow$$

Decomposition Trees in **LI**

We define the notion of decomposable and indecomposable formulas and sequents as follows

Decomposable formula is any formula of the degree ≥ 1

Decomposable sequent is any sequent that contains a decomposable formula

Indecomposable formula is any formula of the degree 0 i.e. is any propositional variable

Decomposition Trees in **LI**

Remark

In a case of formulas written with use of capital letters $A, B, C, ..$ etc , we treat these letters as propositional variables , i.e. as **indecomposable formulas**

Indecomposable sequent is a sequent formed from indecomposable formulas only.

Decomposition Tree Construction (1)

Given a formula A we construct its **decomposition** tree $\mathbf{T_A}$ as follows

Root of the tree $\mathbf{T_A}$ is the sequent $\longrightarrow A$

Given a **node** n of the tree we identify a **decomposition** rule applicable at this node and write its premisses as the **leaves** of the **node** n

We **stop** the decomposition process when we obtain an **axiom** or all leaves of the tree are **indecomposable**

Decomposition Trees in **LI**

Observation 4

The decomposition tree $\mathbf{T_A}$ obtained by the **Construction (1)** most often is not unique

Observation 5

The fact that we **find** a decomposition tree $\mathbf{T_A}$ with a non-axiom leaf **does not** mean that $\nvdash_{LI} A$

This is due to the role of **structural rules** in **LI** and will be discussed later

Proof Search Examples

Examples

We perform proof search and **decide** the existence of proofs in **LI** for a given formula $A \in \mathcal{F}$ by constructing its **decomposition** trees $\mathbf{T_A}$

We examine here some **examples** to show the complexity of the problem

Reminder

In the following and similar examples when building the decomposition trees for formulas representing general schemas we treat the capital letters $A, B, C, D...$ as propositional variables, i.e. as **indecomposable** formulas

Examples

Example 1

Determine] whether

$$\vdash_{\mathbf{LI}} ((\neg A \cap \neg B) \Rightarrow \neg(A \cup B))$$

Observe that

If we find a decomposition tree of A in **LI** such that all its leaves are axiom, we have a proof, i.e

$$\vdash_{\mathbf{LI}} A$$

If all possible decomposition trees have a non-axiom leaf then the proof of A i n **LI** does not exist, i.e.

$$\nvdash_{\mathbf{LI}} A$$

Examples

Consider the following decomposition tree $\mathbf{T1_A}$

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

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

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

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

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

$$\mid (exch \longrightarrow)$$

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

$$\mid (\cap \longrightarrow)$$

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

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

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

$$\mid (\longrightarrow weak)$$

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

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

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

$$\bigwedge (\cup \longrightarrow)$$

$A \longrightarrow B$ $\qquad\qquad$ $B \longrightarrow B$

non – axiom $\qquad\qquad$ *axiom*

Examples

The tree **$T1_A$** has a non-axiom leaf, so it **does not** constitute a proof in **LI**

Observe that the **decomposition** tree in **LI** is not always unique

Hence the existence of a non-axiom leaf **does not** yet prove that the **proof** of A does not exist

Consider the following decomposition tree **$T2_A$**

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

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

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

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

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

$$\mid (exch \longrightarrow)$$

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

$$\mid (\cap \longrightarrow)$$

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

$$\mid (exch \longrightarrow)$$

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

$$\mid (exch \longrightarrow)$$

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

$$\bigwedge (\cup \longrightarrow)$$

$A, \neg A, \neg B \longrightarrow$	$B, \neg A, \neg B \longrightarrow$
$\mid (exch \longrightarrow)$	$\mid (exch \longrightarrow)$
$\neg A, A, \neg B \longrightarrow$	$B, \neg B, \neg A \longrightarrow$
$\mid (\neg \longrightarrow)$	$\mid (exch \longrightarrow)$
$A, \neg B \longrightarrow A$	$\neg B, B, \neg A \longrightarrow$
axiom	$\mid (\neg \longrightarrow)$
	$B, \neg A \longrightarrow B$; *axiom*

Examples

All leaves of $\mathbf{T2_A}$ are axioms

This means that the tree $\mathbf{T2_A}$ is a **a proof** of A in **LI**

We hence proved that

$$\vdash_{\mathbf{LI}} ((\neg A \cap \neg B) \Rightarrow \neg(A \cup B))$$

Examples

Example 2: Show that

1. $\vdash_{\mathbf{LI}} (A \Rightarrow \neg\neg A)$

2. $\nvdash_{\mathbf{LI}} (\neg\neg A \Rightarrow A)$

Solution of **1.**

We construct some, or all **decomposition** trees of

$$\longrightarrow (A \Rightarrow \neg\neg A)$$

A tree $\mathbf{T}_A$ that **ends** with all leaves being axioms is a proof of A in **LI**

Examples

We construct $\mathbf{T}_A$ as follows

$$\longrightarrow (A \Rightarrow \neg\neg A)$$

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

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

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

$$\neg A, A \longrightarrow$$

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

$$A \longrightarrow A$$

axiom

All leaves of $\mathbf{T}_A$ are axioms so we found the **proof**

We **do not** need to construct any other decomposition trees.

Examples

Solution of **2.**

In order to prove that

$$\nvdash_{\mathbf{LI}} \quad (\neg\neg A \Rightarrow A)$$

we have to construct all **decomposition** trees of

$$\longrightarrow (\neg\neg A \Rightarrow A)$$

and show that **each** of them has a non-axiom leaf

Examples

Here is **$T1_A$**

$$\longrightarrow (\neg\neg A \Rightarrow A)$$

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

one of 2 choices

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

$$\mid (\longrightarrow weak)$$

one of 3 choices

$$\neg\neg A \longrightarrow$$

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

one of 3 choices

$$\longrightarrow \neg A$$

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

one of 2 choices

$$A \longrightarrow$$

non – axiom

Here is $\mathbf{T2}_A$

$$\longrightarrow (\neg\neg A \Rightarrow A)$$

$\mid (\longrightarrow\Rightarrow)$ *one of 2 choices*

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

$\mid (contr \longrightarrow)$ *second of 2 choices*

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

$\mid (\longrightarrow weak)$ *first of 2 choices*

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

$\mid (\neg \longrightarrow)$ *first of 2 choices*

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

$\mid (\longrightarrow \neg)$ *one of 2 choices*

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

$\mid (exch \longrightarrow)$ *one of 2 choices*

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

$\mid (\neg \longrightarrow)$*one of 2 choices*

$$A \longrightarrow \neg A$$

$\mid (\longrightarrow \neg)$ *first of 2 choices*

$$A, A \longrightarrow$$

non – axiom

Structural Rules

We can see from the above **decomposition** trees that the "blind" construction of all possible trees only leads to more complicated trees

This is due to the presence of structural rules

The "blind" application of the rule $(contr \rightarrow)$ gives always an infinite number of **decomposition** trees

In order to decide that none of them will produce a proof we need some **extra knowledge** about patterns of their construction, or just simply about the number o useful of application of **structural rules**

Structural Rules

In this case we can just make an ”external” **observation** that the our first tree $\mathbf{T1}_A$ is in a sense a minimal one

It means that all other trees would only **complicate** this one in an inessential way, i.e. the we will never produce a tree with all axioms leaves

One can formulate a deterministic procedure giving a finite number of trees, but the proof of its **correctness** is needed and that requires some extra knowledge

Within the scope of this book we accept the ”external explanation as a **sufficient solution**

Structural Rules

As we can see from the above examples the structural rules and especially the $(contr \longrightarrow)$ rule **complicates** the proof searching task.

Both Gentzen type proof systems **RS** and **GL** from the previous chapter don't contain the structural rules

They also are as we have proved, **complete** with respect to classical semantics.

The original Gentzen system **LK** which does contain the structural rules is also, as proved by Gentzen, **complete**

Structural Rules

Hence all three classical proof system **RS, GL, LK** are equivalent

This proves that the structural rules can be **eliminated** from the system **LK**

A natural question of elimination of **structural rules** from the system **LI** arises

The following **example** illustrates the negative answer

Examples

Example 3

We know that for any formula $A \in \mathcal{F}$,

$$\models A \quad \text{if and only if} \quad \vdash_I \neg\neg A$$

where $\models A$ means that A is classical tautology

$\vdash_I A$ means that A is Intutionistically **provable** in any intuitionistically **complete** proof system $\mathbf{I}$

The system **LI** is intuitionistically **complete** so have that for any formula $A \in \mathcal{F}$,

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

Examples

Obviously $\models (\neg\neg A \Rightarrow A)$, so we must have that

$$\vdash_{\mathbf{LI}} \neg\neg(\neg\neg A \Rightarrow A)$$

We are going to prove now that the rule $(contr \longrightarrow)$ is **essential** to the existence of the proof $\neg\neg(\neg\neg A \Rightarrow A)$

It means that $\neg\neg(\neg\neg A \Rightarrow A)$ **is not provable** without the rule $(contr \longrightarrow)$

The following decomposition tree $\mathbf{T}_A$ is a proof of $\neg\neg(\neg\neg A \Rightarrow A)$ **with use** of the rule $(contr \longrightarrow)$

Examples

$$\longrightarrow \neg\neg(\neg\neg A \Rightarrow A)$$

$\mid (\longrightarrow \neg)$

$$\neg(\neg\neg A \Rightarrow A) \longrightarrow$$

$\mid (\mathit{contr} \longrightarrow)$

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

$\mid (\neg \longrightarrow)$

$$\neg(\neg\neg A \Rightarrow A) \longrightarrow (\neg\neg A \Rightarrow A)$$

$\mid (\longrightarrow\Rightarrow)$

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

$\mid (\longrightarrow \mathit{weak})$

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

$\mid (\neg \longrightarrow)$

$$\neg(\neg\neg A \Rightarrow A) \longrightarrow \neg A$$

$\mid (\longrightarrow \neg)$

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

$\mid (\mathit{exch} \longrightarrow)$

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

$\mid (\neg \longrightarrow)$

$$A \longrightarrow (\neg\neg A \Rightarrow A)$$

$\mid (\longrightarrow\Rightarrow)$

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

axiom

Contraction Rule

Assume now that the rule $(contr \longrightarrow)$ **is not** available. All possible decomposition trees are as follows

Tree **T1**$_A$

$$\longrightarrow \neg\neg(\neg\neg A \Rightarrow A)$$

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

$$\neg(\neg\neg A \Rightarrow A) \longrightarrow$$

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

$$\longrightarrow (\neg\neg A \Rightarrow A)$$

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

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

$$\mid (\longrightarrow weak)$$

$$\neg\neg A \longrightarrow$$

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

$$\longrightarrow \neg A$$

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

$$A \longrightarrow$$

non – axiom

Contraction Rule

The next is $\mathbf{T2}_A$

$$\longrightarrow \neg\neg(\neg\neg A \Rightarrow A)$$

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

$$\neg(\neg\neg A \Rightarrow A) \longrightarrow$$

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

$$\longrightarrow (\neg\neg A \Rightarrow A)$$

$$\mid (\longrightarrow weak)$$

$$\longrightarrow$$

$$non - axiom$$

Contraction Rule

The next is $\mathbf{T3}_A$

$$\longrightarrow \neg\neg(\neg\neg A \Rightarrow A)$$

$$\mid (\longrightarrow weak)$$

$$\longrightarrow$$

$$non - axiom$$

Contraction Rule

The last one is **T4$_A$**

$$\longrightarrow \neg\neg(\neg\neg A \Rightarrow A)$$

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

$$\neg(\neg\neg A \Rightarrow A) \longrightarrow$$

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

$$\longrightarrow (\neg\neg A \Rightarrow A)$$

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

$$]$$

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

$$\mid (\longrightarrow weak)$$

$$\neg\neg A \longrightarrow$$

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

$$\longrightarrow \neg A$$

$$\mid (\longrightarrow weak)$$

$$\longrightarrow$$

non – axiom

Contraction Rule

We have considered all possible decomposition trees that **do not** involve the contraction rule $(contr \longrightarrow)$ and **none** of them was a proof

This shows that the formula

$$\neg\neg(\neg\neg A \Rightarrow A)$$

is not provable in **LI** without $(contr \longrightarrow)$ rule, i.e. that we proved the following

Fact

The contraction rule $(contr \longrightarrow)$ **can not** be eliminated from **LI**

Proof Search Heuristic Method

Proof Search Heuristic Method

Before we define a heuristic method of searching for proof in **LI** let's make some additional observations to the already made **observations 1-5**

Observation 6

The goal of constructing the decomposition tree is to **obtain** axioms or indecomposable leaves

With respect to this goal the use **logical** decomposition rules has a priority over the use of the **structural** rules

We use this information while describing the proof search **heuristic**

Proof Search Heuristic Method

Observation 7

All logical decomposition rules $(\circ \rightarrow)$, where $\circ$ denotes any connective, must have a formula we want to decompose as the first formula at the decomposition node

It means that if we want to **decompose** a formula $\circ A$ the node must have a form $\circ A, \Gamma \longrightarrow \Delta$

Remember: order of decomposition is important

Also sometimes it is necessary to decompose a **formula within the sequence** Γ **first**, before decomposing $\circ A$ in order to **find** a proof

Proof Search Heuristic Method

For example, consider two nodes

$$n_1 = \neg\neg A,\ (A \cap B) \longrightarrow B$$

and

$$n_2 = (A \cap B),\ \neg\neg A \longrightarrow B$$

We are going to see that the results of decomposing n_1 and n_2 **differ dramatically**

Let's decompose the node n_1

Observe that the only way to be able to decompose the formula $\neg\neg A$ is to use the rule $(\rightarrow \mathit{weak})$ as a **first step**

The **two possible** decomposition trees that starts at the node n_1 are as follows

Proof Search Heuristic Method

First Tree

T1$_{n_1}$

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

$$\mid (\rightarrow weak)$$

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

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

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

$$\mid (\cap \rightarrow)$$

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

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

$$A, A, B \longrightarrow$$

non – axiom

Proof Search Heuristic Method

Second Tree

$\mathbf{T2}_{n_1}$

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

$$\mid (\to weak)$$

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

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

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

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

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

$$\mid (\cap \to)$$

$$A, A, B \longrightarrow$$

non – axiom

Proof Search Heuristic Method

Let's now decompose the node n_2

Observe that following our **Observation 6** we start by decomposing the formula $(A \cap B)$ by the use of the rule $(\cap \rightarrow)$ as the **first step**

A decomposition tree that starts at the node n_2 is as follows

$$\mathbf{T}_{n_2}$$

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

$$\mid (\cap \rightarrow)$$

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

$$\textit{axiom}$$

This proves that the node n_2 is **provable** in **LI**, i.e.

$$\vdash_{\mathbf{LI}} \quad (A \cap B), \neg\neg A \longrightarrow B$$

Proof Search Heuristic Method

Observation 8

The use of structural rules is **important** and **necessary** while we search for proofs

Nevertheless we have to **use them** on the "must" basis and set up some **guidelines** and **priorities** for their use

For example, the use of weakening rule **discharges** the weakening formula, and hence we might **loose an information** that may be essential to finding the **proof**

We should use the weakening rule only when it is **absolutely necessary** for the next decomposition steps

Proof Search Heuristic Method

Hence, the use of weakening rule $(\rightarrow weak)$ **can**, and **should be restricted** to the cases when it leads to **possibility** of the future use of the **negation rule** $(\neg \rightarrow)$

This was the case of the decomposition tree $\mathbf{T1}_{n_1}$

We used the rule $(\rightarrow weak)$ as an necessary step, but it **discharged** too much information and we didn't get a proof, when **proof on this node existed**

Proof Search Heuristic Method

Here is such a proof

T3$_{n_1}$

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

$$\mid (exch \longrightarrow)$$

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

$$\mid (\cap \rightarrow)$$

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

axiom

Method

For any $A \in \mathcal{F}$ we construct the set of decomposition trees $\mathbf{T}_{\rightarrow A}$ following the rules below.

1. Use first logical rules where applicable.

2. Use $(exch \rightarrow)$ rule to decompose, via logical rules, as many formulas on the left side of $\longrightarrow$ as possible

Remember that the order of decomposition matters! so you have to cover different choices

3. Use $(\rightarrow weak)$ only on a "must" basis and in connection with the **possibility** of the future use of the $(\neg \rightarrow)$ rule

4. Use $(contr \rightarrow)$ rule as the **last recourse** and only to formulas that contain $\neg$ or $\Rightarrow$ as a main connective

5. Let's call a formula A to which we apply $(contr \rightarrow)$ rule a **a contraction formula**

6. The only contraction formulas are formulas containing $\neg$ or $\Rightarrow$ between theirs logical connectives

Proof Search Heuristic Method

7. Within the process of construction of all possible trees use $(contr \rightarrow)$ rule **only** to **contraction formulas**

8. Let C be a **contraction formula** appearing on a node n of the decomposition tree of $\mathbf{T}_{\rightarrow A}$

For any **contraction formula** C, any node n, we apply $(contr \rightarrow)$ rule to the the formula C at the node n **at most** as many times as the number of sub-formulas of C

If we **find** a tree with all axiom leaves we have a **proof**, i.e.

$$\vdash_{LI} A$$

If **all trees** (finite number) have a non-axiom leaf we have proved that proof of A **does not exist**, i.e.

$$\nvdash_{LI} A$$