cse371/math371
LOGIC

Professor Anita Wasilewska

LECTURE 6a

Chapter 6
Automated Proof Systems
Completeness of Classical Propositional Logic

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

Strong Soundness and Constructive Completeness

Gentzen Sequent Systems **GL, G**

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

Their axioms are, and the rules of inference **operate** on expressions called by Gentzen **sequents**
Hence the name Gentzen Sequent Systems

The Gentzen **original** system **LK** is presented and discussed in detail in the next Lecture 6b

Gentzen Sequent System **GL**

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

Both proof systems **GL** and **G** admit 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**

We define **GL** components are as follows

Language

We adopt a propositional language

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

and we add to it a new symbol $\longrightarrow$ called a **Gentzen arrow**

It means we consider formally a new language

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

Gentzen Sequent System **GL**

Sequents

The **sequents** are expressions built out of finite sequences (empty included) of formulas of the language $\mathcal{L}_{\{\cup,\cap,\Rightarrow,\neg\}}$ 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 defined as

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

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 $\Gamma', \Delta' \in \mathbf{VAR}^*$

Indecomposable Sequents

Remark

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

$$\Gamma', \Delta', \ldots\ldots$$

denote sequences of **positive literals** i.e. only **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 $\Gamma'_1, \Gamma'_2, \Delta'_1, \Delta'_2 \in \mathbf{VAR}^*$

GL Components: Axioms

Semantic Link

Consider axiom

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

We evaluate, for any truth assignment $v : VAR \longrightarrow \{T, F\}$

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

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

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

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

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

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

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

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

GL Rules

Negation rules

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

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

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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$a \longrightarrow b, a$$

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

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

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

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

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

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

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

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

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

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

$\longrightarrow \ a, b, a$ — *non – axiom*

$b \longrightarrow b, a$ — *axiom*

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

We have one more tree to construct

System **GL** Exercises

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

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

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

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

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

$\longrightarrow \ (\neg b \Rightarrow a), a$	$b \longrightarrow \ (\neg b \Rightarrow a)$
$\mid (\longrightarrow\Rightarrow)$ (*one choice*)	$\mid (\rightarrow\Rightarrow)$ (*one choice*)
$\neg b \longrightarrow \ a, a$	$b, \neg b \longrightarrow \ a$
$\mid (\neg \rightarrow)$ (*one choice*)	$\mid (\neg \rightarrow)$ (*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

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

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

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

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

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

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

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

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

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

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

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

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

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

$\longrightarrow \neg a, b, a$ | $b \longrightarrow b, a$

$\mid (\rightarrow \neg)$ | *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 i nference 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

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

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

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

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

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

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

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

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

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

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

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

$\longrightarrow b, b, a$ **F**	$a \longrightarrow b, a$
non – axiom	*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

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 **GL**

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

Axioms of **G** are

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

where

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

Gentzen Proof System **G**

Rules of Inference

Conjunction

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

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

Disjunction

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

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

Gentzen Proof System **G**

Implication

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

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

Negation

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

where

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

System **G** Exercises

Exercises

Follow the example of the **GL** system and adopt all needed definitions and proofs to prove the **completeness** of the proof 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 d** ecomposition trees $\mathbf{T}_{\Gamma \rightarrow \Delta}$ are **finite**

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

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 $\Gamma \longrightarrow \Delta \in SQ$,

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