cse371/Math371
LOGIC

Professor Anita Wasilewska
LECTURE 6c
Chapter 5, Chapter 6
REVIEW for Q2

Chapter 5: System $H_2$ and examples of formal proofs in $H_2$

Chapter 6: Proof Systems RS, RS1, RS2

Chapter 6: Proof Systems GL, G
Definition

\( H_2 = ( L_{\rightarrow, \neg}, \mathcal{F}, \{ A1, A2, A3 \} (MP) ) \)

A1 (Law of simplification)
\( (A \rightarrow (B \rightarrow A)) \)

A2 (Frege’s Law)
\( (((A \rightarrow (B \rightarrow C))) \rightarrow (((A \rightarrow B) \rightarrow (A \rightarrow C)))) \)

A3 \( (((\neg B \rightarrow \neg A) \rightarrow (((\neg B \rightarrow A) \rightarrow B))) \)

MP (Rule of inference)

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

where \( A, B, C \) are any formulas of the propositional language \( L_{\rightarrow, \neg} \)
Deduction Theorem for $H_2$

For any $A, B \in \mathcal{F}$ and $\Gamma \subseteq \mathcal{F}$

$$\Gamma, A \vdash_{H_1} B \quad \text{if and only if} \quad \Gamma \vdash_{H_2} (A \Rightarrow B)$$

In particular

$$A \vdash_{H_2} B \quad \text{if and only if} \quad \vdash_{H_2} (A \Rightarrow B)$$
Formal Proofs

The proof of the following Lemma provides a good example of multiple applications of the Deduction Theorem

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

(a) $(A \Rightarrow B), (B \Rightarrow C) \vdash_{H_2} (A \Rightarrow C),$

(b) $(A \Rightarrow (B \Rightarrow C)) \vdash_{H_2} (B \Rightarrow (A \Rightarrow C))$

Observe that by Deduction Theorem we can re-write (a) as

(a') $(A \Rightarrow B), (B \Rightarrow C), A \vdash_{H_2} C$
Soundness and Completeness Theorems

We get by easy verification

**Soundness Theorem** \( H_2 \)
For every formula \( A \in \mathcal{F} \)

\[
\text{if } \vdash_{H_2} A \text{ then } \models A
\]

We prove in the next Lecture, that \( H_2 \) is also complete, i.e. we prove

**Completeness Theorem** for \( H_2 \)
For every formula \( A \in \mathcal{F} \),

\[
\vdash_{H_2} A \text{ if and only if } \models A
\]
FORMAL PROOFS IN $H_2$
Examples and Exercises

We present now some examples of formal proofs in $H_2$
There are two reasons for presenting them.

**First reason** is that all formulas we prove here to be provable play a crucial role in the proof of Completeness Theorem for $H_2$

**The second reason** is that they provide a "training ground" for a reader to learn how to develop formal proofs.

For this reason we write some proofs in a full detail and we leave some for the reader to complete in a way explained in the following example.
Important Lemma

We write $\vdash$ instead of $\vdash_{H_2}$ for the sake of simplicity.

Reminder

In the construction of the formal proofs we often use the Deduction Theorem and the following Lemma 1 they was proved in previous section

Lemma 1

(a) $(A \Rightarrow B), (B \Rightarrow C) \vdash_{H_2} (A \Rightarrow C)$

(b) $(A \Rightarrow (B \Rightarrow C)) \vdash_{H_2} ((B \Rightarrow (A \Rightarrow C))$
Example 1

Example 1

Here are consecutive steps

$$B_1, \ldots, B_5, B_6$$

of the proof in $H_2$ of $(\neg\neg B \Rightarrow B)$

$B_1 : \quad ((\neg B \Rightarrow \neg\neg B) \Rightarrow ((\neg B \Rightarrow \neg B) \Rightarrow B))$

$B_2 : \quad ((\neg B \Rightarrow \neg B) \Rightarrow ((\neg B \Rightarrow \neg\neg B) \Rightarrow B))$

$B_3 : \quad (\neg B \Rightarrow \neg B)$

$B_4 : \quad ((\neg B \Rightarrow \neg\neg B) \Rightarrow B)$

$B_5 : \quad (\neg\neg B \Rightarrow (\neg B \Rightarrow \neg\neg B))$

$B_6 : \quad (\neg\neg B \Rightarrow B)$
Exercise 1

Exercise 1
Complete the proof presented in Example 1 by providing comments how each step of the proof was obtained.

ATTENTION

The solution presented on the next slide shows you how you will have to write details of your solutions on the TESTS Solutions of other problems presented later are less detailed. Use them as exercises to write a detailed, complete solutions.
Exercise 1 Solution

Solution

The comments that complete the proof are as follows.

$B_1 : ((\neg B \Rightarrow \neg \neg B) \Rightarrow ((\neg B \Rightarrow \neg B) \Rightarrow B))$

Axiom A3 for $A = \neg B, B = B$

$B_2 : ((\neg B \Rightarrow \neg B) \Rightarrow ((\neg B \Rightarrow \neg \neg B) \Rightarrow B))$

$B_1$ and Lemma 1 (b) for

$A = (\neg B \Rightarrow \neg \neg B), B = (\neg B \Rightarrow \neg B), C = B$, i.e. we have

$[((\neg B \Rightarrow \neg \neg B) \Rightarrow ((\neg B \Rightarrow \neg B) \Rightarrow B)) \vdash ((\neg B \Rightarrow \neg B) \Rightarrow ((\neg B \Rightarrow \neg \neg B) \Rightarrow B))]$
Exercise 1 Solution

$B_3 : (\neg B \Rightarrow \neg B)$
We proved for $H_1$ and hence for $H_2$ that $\vdash (A \Rightarrow A)$ and we substitute $A = \neg B$

$B_4 : ((\neg B \Rightarrow \neg \neg B) \Rightarrow B)$
$B_2, B_3$ and MP

$B_5 : (\neg \neg B \Rightarrow (\neg B \Rightarrow \neg \neg B))$
Axiom A1 for $A = \neg \neg B, B = \neg B$

$B_6 : (\neg \neg B \Rightarrow B)$
$B_4, B_5$ and Lemma 1 (a) for
$A = \neg \neg B, B = (\neg B \Rightarrow \neg \neg B), C = B$; i.e.
$(\neg \neg B \Rightarrow (\neg B \Rightarrow \neg \neg B)), ((\neg B \Rightarrow \neg \neg B) \Rightarrow B) \vdash (\neg \neg B \Rightarrow B)$
Proofs from Axioms Only

General remark

Observe that in steps $B_2, B_3, B_5, B_6$ we call on previously proved facts and use them as a part of our proof.

We can obtain a proof that uses only axioms by inserting previously constructed formal proofs of these facts into the places occupying by the steps $B_2, B_3, B_5, B_6$.

For example in previously constructed proof of $(A \Rightarrow A)$ we replace $A$ by $\neg B$ and insert such constructed proof of $(\neg B \Rightarrow \neg B)$ after step $B_2$.

The last step of the inserted proof becomes now ”old” step $B_3$ and we re-numerate all other steps accordingly.
Proofs from Axioms Only

Here are consecutive first THREE steps of the proof of \((\neg\neg B \Rightarrow B)\)

\(B_1:\) \[((\neg B \Rightarrow \neg\neg B) \Rightarrow ((\neg B \Rightarrow \neg B) \Rightarrow B))\]

\(B_2:\) \[((\neg B \Rightarrow \neg B) \Rightarrow ((\neg B \Rightarrow \neg\neg B) \Rightarrow B))\]

\(B_3:\) \((\neg B \Rightarrow \neg B)\)

We insert now the proof of \((\neg B \Rightarrow \neg B)\) after step \(B_2\) and erase the \(B_3\)

The last step of the inserted proof becomes the erased \(B_3\)

Proofs from Axioms Only

A part of new \textbf{transformed} proof is

$B_1 : ((\neg B \Rightarrow \neg \neg B) \Rightarrow ((\neg B \Rightarrow \neg B) \Rightarrow B))$ \ (Old \ $B_1$)

$B_2 : ((\neg B \Rightarrow \neg B) \Rightarrow ((\neg B \Rightarrow \neg \neg B) \Rightarrow B))$ \ (Old \ $B_2$)

We insert here the proof from axioms only of Old $B_3$

$B_3 : ((\neg B \Rightarrow ((\neg B \Rightarrow \neg B) \Rightarrow \neg B))) \Rightarrow ((\neg B \Rightarrow (\neg B \Rightarrow \neg B))) \Rightarrow (\neg B \Rightarrow \neg B))$, \ (New \ $B_3$)

$B_4 : (\neg B \Rightarrow ((\neg B \Rightarrow \neg B) \Rightarrow \neg B))$

$B_5 : ((\neg B \Rightarrow (\neg B \Rightarrow \neg B)) \Rightarrow (\neg B \Rightarrow \neg B)))$

$B_6 : (\neg B \Rightarrow (\neg B \Rightarrow \neg B))$

$B_7 : (\neg B \Rightarrow \neg B)$ \ (Old \ $B_3$)
Proofs from Axioms Only

We repeat our procedure by replacing the step $B_2$ by its formal proof as defined in the proof of the Lemma 1 (b).

We continue the process for all other steps which involved application of the Lemma 1 until we get a full formal proof from the axioms of $H_2$ only.

Usually we don’t do it and we don’t need to do it, but it is important to remember that it always can be done.
Example 2

Here are consecutive steps

\[ B_1, B_2, \ldots, B_5 \]

in a proof of \((B \Rightarrow \neg \neg B)\)

- \(B_1\) \((\neg \neg \neg B \Rightarrow \neg B) \Rightarrow ((\neg \neg \neg B \Rightarrow B) \Rightarrow \neg \neg B))\)
- \(B_2\) \((\neg \neg \neg B \Rightarrow \neg B)\)
- \(B_3\) \((\neg \neg \neg B \Rightarrow B) \Rightarrow \neg \neg B)\)
- \(B_4\) \((B \Rightarrow (\neg \neg \neg B \Rightarrow B))\)
- \(B_5\) \((B \Rightarrow \neg \neg B)\)
Exercise 2

Complete the proof presented in Example 2 by providing detailed comments how each step of the proof was obtained.

Solution

The comments that complete the proof are as follows.

$B_1 \quad (\neg\neg\neg B \Rightarrow \neg B) \Rightarrow ((\neg\neg\neg B \Rightarrow B) \Rightarrow \neg\neg B)$

Axiom A3 for $A = B, B = \neg\neg B$

$B_2 \quad (\neg\neg\neg B \Rightarrow \neg B)$

Example 1 for $B = \neg B$
Exercise 2

\[ B_3 \ ( ( \neg \neg \neg B \Rightarrow B ) \Rightarrow \neg \neg B ) \]

\( B_1, B_2 \) and \( \text{MP} \), i.e.

\[
( \neg \neg \neg B \Rightarrow \neg B ) ; ( ( \neg \neg \neg B \Rightarrow \neg B ) \Rightarrow ( ( \neg \neg \neg B \Rightarrow B ) \Rightarrow \neg \neg B ) )
\]

\[
( ( \neg \neg \neg B \Rightarrow B ) \Rightarrow \neg \neg B )
\]

\[ B_4 \ ( B \Rightarrow ( \neg \neg \neg B \Rightarrow B ) ) \]

Axiom \( A1 \) for \( A = B, \ B = \neg \neg B \)

\[ B_5 \ ( B \Rightarrow \neg \neg B ) \]

\( B_3, B_4 \) and lemma 1a for \( A = B, B = ( \neg \neg \neg B \Rightarrow B ), C = \neg \neg B \),

i.e.

\[
( B \Rightarrow ( \neg \neg \neg B \Rightarrow B ) ) , ( ( \neg \neg \neg B \Rightarrow B ) \Rightarrow \neg \neg B ) \vdash ( B \Rightarrow \neg \neg B )
\]
CHAPTER 6
RS Proof Systems
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 $T_A$, called a decomposition tree.

Building a decomposition tree what really is a proof search tree consists in the first step of transforming the RS rules into corresponding decomposition rules.
Tree Rules

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

\[
\Gamma', (A \cup B), \Delta
\]

\[
\vdash (\cup)
\]

\[
\Gamma', A, B, \Delta
\]
Tree Rules

(¬∪) rule

Γ′, ¬(A ∪ B), Δ

∧(¬∪)

Γ′, ¬A, Δ  Γ′, ¬B, Δ

(∩) rule

Γ′, (A ∩ B), Δ

∧(∩)

Γ′, A, Δ  Γ′, B, Δ
Tree Rules

\((\neg \cup)\) rule

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

\(\vdash (\neg \cap)\)

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

\((\Rightarrow)\) rule

\(\Gamma', (A \Rightarrow B), \Delta\)

\(\vdash (\Rightarrow)\)

\(\Gamma', \neg A, B, \Delta\)
Tree Rules

(\neg \Rightarrow) \text{ rule}

\Gamma', \neg (A \Rightarrow B), \Delta

\bigwedge (\neg \Rightarrow)

\Gamma', A, \Delta \quad \Gamma', \neg B, \Delta

(\neg\neg) \text{ rule}

\Gamma', \neg\neg A, \Delta

| (\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 the decomposition tree 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 $\Gamma'$ built only out of literals, i.e. $\Gamma \in \mathcal{F}'^{*}$ is called an \textbf{indecomposable sequence}.

Definition
A formula $A$ that is not a literal, i.e. $A \in \mathcal{F} - \mathcal{L}T$ is called a \textbf{decomposable formula}.

Definition
A sequence $\Gamma$ that contains a \textbf{decomposable formula} is called a \textbf{decomposable sequence}.
Definitions and Observations

Observation 1
For any decomposable sequence, i.e. for any $\Gamma \not\in LT^*$ there is exactly one decomposition rule that can be applied to it.

This rule is determined by the first decomposable formula in $\Gamma$ and by the main connective of that formula.
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 $\Gamma \in F^*$, $\Gamma \in LT^*$ or $\Gamma$ is in the domain of exactly one of RS Decomposition Rules
This rule is determined by the first decomposable formula in $\Gamma$ and by the main connective of that formula
Decomposition Tree Definition

Definition: Decomposition Tree $T_A$
For each $A \in \mathcal{F}$, a decomposition tree $T_A$ is a tree built as follows

Step 1.
The formula $A$ is the root of $T_A$
For any other node $\Gamma$ of the tree we follow the steps below

Step 2.
If $\Gamma$ is indecomposable then $\Gamma$ becomes a leaf of the tree
Decomposition Tree Definition

Step 3.
If \( \Gamma \) is decomposable, then we traverse \( \Gamma \) 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 $\Gamma \in F^*$ the following conditions hold
1. $T_{\Gamma}$ is finite and unique
2. $T_{\Gamma}$ is a proof of $\Gamma$ in RS if and only if all its leafs are axioms
3. $\not\proves_{RS} \Gamma$ if and only if $T_{\Gamma}$ has a non-axiom leaf
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** \( A \in \mathcal{F} \)

\[ \vdash_{RS} A \text{ if and only if } \models A \]

and then we generalize it to the following:

**Completeness Theorem 2** \( \Gamma \in \mathcal{F}^* \),

\[ \vdash_{RS} \Gamma \text{ if and only if } \models \Gamma \]

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

\[ S = (L, E, LA, R) \]

Definition
A rule \( r \in R \) 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 \( S \) is sound and all its rules \( r \in R \) 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 \((\cup)\)

\[
\frac{\Gamma', A, B, \Delta}{\Gamma', (A \cup B), \Delta}
\]

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

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 $T_A$ 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 formula $A$

This means that any $v$ that falsifies a non-axiom leaf is a counter-model for $A$
Counter Model Theorem

We have proved the following

Counter Model Theorem
Let $A \in \mathcal{F}$ be such that its decomposition tree $T_A$ 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 $T_A$
Consider a tree \( T_A \)

\[
(((a \Rightarrow b) \cap \lnot c) \cup (a \Rightarrow c))
\]

\[
\big| (\cup)
\]

\[
((a \Rightarrow b) \cap \lnot c), (a \Rightarrow c)
\]

\[
\bigwedge (\cap)
\]

\[
(a \Rightarrow b), (a \Rightarrow c)
\]

\[
\big| (\Rightarrow)
\]

\[
\lnot a, b, (a \Rightarrow c)
\]

\[
\big| (\Rightarrow)
\]

\[
\lnot a, b, \lnot a, c
\]

\[
\lnot c, (a \Rightarrow c)
\]

\[
\big| (\Rightarrow)
\]

\[
\lnot c, \lnot a, c
\]

\[
\lnot a, b, \lnot a, c
\]
Counter Model Example

The tree $T_A$ has a non-axiom leaf

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

We want to define a truth assignment $v : \text{VAR} \rightarrow \{T, F\}$ 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 \( T_A \)

\[
(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c)) = F
\]

\[
| (\cup)
\]

\[
((a \Rightarrow b) \cap \neg c), (a \Rightarrow c) = F
\]

\[
\cap (\cap)
\]

\[
(a \Rightarrow b), (a \Rightarrow c) = F
\]

\[
| (\Rightarrow)
\]

\[
\neg a, b, (a \Rightarrow c) = F
\]

\[
| (\Rightarrow)
\]

\[
\neg a, b, \neg a, c = F
\]
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 $T_A$.

In our case the formula $T_A$ has only **one non-axiom leaf**, and hence only one restricted **counter model**.
RS Completeness Theorem

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

If $\models A$, then $\vdash_{RS} A$

We prove instead the opposite implication

RS Completeness Theorem

If $\not\vdash_{RS} A$ then $\not\models A$
Proof of Completeness Theorem

Proof of Completeness Theorem
Assume that $A$ is any formula is such that 

$$\not\models_{RS} A$$

By the Decomposition Tree Theorem the $T_A$ 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.

$$\not\models A$$
System **RS2 Definition**

**RS2 Definition**
System **RS2** is a proof system obtained from **RS** by changing the sequences $\Gamma'$ into $\Gamma$ 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

\[(\neg(\neg a \Rightarrow (a \land \neg b)) \Rightarrow (\neg a \land (\neg a \lor \neg b)))\]
RS2 Exercises

**T1_A**

\[ \neg(\neg a \Rightarrow (a \land \neg b)) \Rightarrow (\neg a \land (\neg a \lor \neg b)) \]

\[ \models (\Rightarrow) \]

\[ \neg\neg(\neg a \Rightarrow (a \land \neg b)), (\neg a \land (\neg a \lor \neg b)) \]

\[ \models (\neg\neg) \]

\[ (\neg a \Rightarrow (a \land \neg b)), (\neg a \land (\neg a \lor \neg b)) \]

\[ \models (\Rightarrow) \]

\[ \neg\neg a, (a \land \neg b), (\neg a \land (\neg a \lor \neg b)) \]

\[ \models (\neg\neg) \]

\[ a, (a \land \neg b), (\neg a \land (\neg a \lor \neg b)) \]

\[ \land (\cap) \]

\[ a, a, (\neg a \land (\neg a \lor \neg b)) \]

\[ a, \neg b, (\neg a \land (\neg a \lor \neg b)) \]

\[ \land (\cap) \]

\[ a, a, \neg a, (\neg a \lor \neg b) \]

\[ a, a, (\neg a \lor \neg b) \]

\[ a, \neg b, \neg a \]

\[ \models (\lor) \]

\[ a, a, \neg a, \neg a, \neg b \]

\[ a, a, \neg a, \neg b \]

\[ \text{axiom} \]

\[ a, \neg b, (\neg a \lor \neg b) \]

\[ \models (\lor) \]

\[ a, \neg b, \neg a, \neg b \]

\[ \text{axiom} \]
RS2 Exercises

$T_{2A}$

\[ (\neg(\neg a \Rightarrow (a \cap \neg b)) \Rightarrow (\neg a \cap (\neg a \cup \neg b))) \]

$| (\Rightarrow) \]

\[ \neg(\neg a \Rightarrow (a \cap \neg b)), (\neg a \cap (\neg a \cup \neg b)) \]

$| (\neg) \]

\[ (\neg a \Rightarrow (a \cap \neg b)), (\neg a \cap (\neg a \cup \neg b)) \]

$\bigwedge(\cap)$

\[ (\neg a \Rightarrow (a \cap \neg b)), \neg a \]

$| (\Rightarrow) \]

\[ \neg a, (a \cap \neg b), \neg a \]

$| (\neg) \]

\[ a, (a \cap \neg b), \neg a \]

$\bigwedge(\cap)$

\[ a, a, \neg a \]

axiom

\[ a, \neg b, \neg a \]

axiom

\[ a, a, \neg a \]

axiom

\[ a, \neg b, \neg a \]

axiom

\[ a, a, \neg a, \neg b \]

axiom

\[ a, \neg b, \neg a, \neg b \]

axiom
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
Exercise 3
Define shortly, in your own words, for any formula $A$, its decomposition tree $T_A$ 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 $T_A$ can be defined as follows
It has the formula 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.
if 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. there is no decomposition rule 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 $\Gamma$ instead of $\Gamma'$ 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

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

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

This leads to existence of non-unique trees.
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,

\[ \kappa_{RS2} A \quad \text{then} \quad \not\models A \]

Assume \( \kappa_{RS2} A \)
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 $\nu$ which falsifies $A$, as follows:

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

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.

$$
\not\models A
$$
CHAPTER 6
Gentzen GL Proof Systems
Gentzen System **GL** Definition

**Definition**

\[ \text{GL} = ( L\{\cup, \cap, \Rightarrow, \neg\}, \ SQ, \ LA, \ R ) \]

where

\[ SQ = \{ \Gamma \rightarrow \Delta : \Gamma, \Delta \in F^* \} \]

\[ R = \{(\cap \rightarrow), (\rightarrow \cap), (\cup \rightarrow), (\rightarrow \cup), (\Rightarrow \rightarrow), (\rightarrow \Rightarrow)\} \]

\[ \cup \{(\neg \rightarrow), (\rightarrow \neg)\} \]

We write, as usual,

\[ \Vdash_{\text{GL}} \Gamma \rightarrow \Delta \]

to denote that \( \Gamma \rightarrow \Delta \) has a formal proof in \( \text{GL} \)

For any formula \( A \in F \)

\[ \Vdash_{\text{GL}} A \quad \text{if ad only if} \quad \rightarrow A \]

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 $\Gamma \rightarrow \Delta$ is a tree $T_{\Gamma \rightarrow \Delta}$

of sequents satisfying the following conditions:

1. The topmost sequent, i.e. the root of $T_{\Gamma \rightarrow \Delta}$ is $\Gamma \rightarrow \Delta$

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 \textbf{GL} of the de Morgan Law

\[\rightarrow (\neg(a \land b) \Rightarrow (\neg a \lor \neg b))\]
\[\mid (\rightarrow \Rightarrow)\]
\[\neg(a \land b) \rightarrow (\neg a \lor \neg b)\]
\[\mid (\rightarrow \lor)\]
\[\neg(a \land b) \rightarrow \neg a, \neg b\]
\[\mid (\rightarrow \neg)\]
\[b, \neg(a \land b) \rightarrow \neg a\]
\[\mid (\rightarrow \neg)\]
\[b, a, \neg(a \land b) \rightarrow\]
\[\mid (\neg \rightarrow)\]
\[b, a \rightarrow (a \land b)\]
\[\land (\rightarrow \land)\]

\[b, a \rightarrow a\] \hspace{1cm} \[b, a \rightarrow b\]
Example

Here is another tree-proof in $\textbf{GL}$ of the de Morgan Law

\[
\rightarrow (\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))
\]

\[
\begin{align*}
| (\rightarrow \Rightarrow) \\
\neg(a \cap b) & \rightarrow (\neg a \cup \neg b) \\
| (\rightarrow \lor) \\
\neg(a \cap b) & \rightarrow \neg a, \neg b \\
| (\rightarrow \neg) \\
b, \neg(a \cap b) & \rightarrow \neg a \\
| (\neg \rightarrow) \\
b & \rightarrow \neg a, (a \cap b) \\
\end{align*}
\]

\[
\begin{align*}
\land (\rightarrow \land) \\
b & \rightarrow \neg a, a & b & \rightarrow \neg a, b \\
| (\rightarrow \neg) \\
b, a & \rightarrow a & b, a & \rightarrow b \\
\end{align*}
\]
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 a formula $A$ sequent $\rightarrow A$.

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 $\text{GL}$.

If all trees in $\mathcal{T}_A$ have at least one non-axiom leaf, the proof of $A$ does not exist.
Exercise
Prove, by constructing proper decomposition trees that

$$\neg_{GL} ((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))$$

Solution
For some formulas $A$, their decomposition tree $T_{\rightarrow A}$ 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 $\rightarrow A$ as follows
System GL Exercises

\[ T_1 \rightarrow A \]

\[ \rightarrow ((a \rightarrow b) \Rightarrow (\neg b \Rightarrow a)) \]

\[ | (\rightarrow\Rightarrow) \ (one\ choice) \]

\[ (a \Rightarrow b) \rightarrow (\neg b \Rightarrow a) \]

\[ | (\rightarrow\Rightarrow) \ (first\ of\ two\ choices) \]

\[ \neg b, (a \Rightarrow b) \rightarrow a \]

\[ | (\neg \rightarrow) \ (one\ choice) \]

\[ (a \Rightarrow b) \rightarrow b, a \]

\[ \wedge (\Rightarrow\Rightarrow) \ (one\ choice) \]

\[ \rightarrow a, b, a \]

\[ b \rightarrow b, a \]

\textit{non – axiom} \hspace{1cm} \textit{axiom}

The tree contains a \textbf{non-axiom} leaf, hence it is \textbf{not a proof}

We have \textbf{one more tree} to construct
System **GL** Exercises

**T**\(_2\) \(\rightarrow\) \(A\)

\(\rightarrow ((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))\)

\(\vdash (\leftrightarrow) \text{ (one choice)}\)

\(a \Rightarrow b \rightarrow (\neg b \Rightarrow a)\)

\(\land (\Rightarrow \rightarrow) \text{ (second choice)}\)

\(\rightarrow (\neg b \Rightarrow a), a\)

\(b \rightarrow (\neg b \Rightarrow a)\)

\(\vdash (\leftrightarrow \rightarrow) \text{ (one choice)}\)

\(\neg b \rightarrow a, a\)

\(b, \neg b \rightarrow a\)

\(\vdash (\neg \rightarrow) \text{ (one choice)}\)

\(\rightarrow b, a, a\)

\(b \rightarrow b, a\)

non-axiom

axiom

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

\(\kappa_{\text{GL}} ((a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a))\)
Does the tree below constitute a proof in GL? Justify your answer.

\[
\begin{align*}
T \rightarrow A \\
\rightarrow \neg \neg ((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a)) \\
\mid (\rightarrow \neg) \\
\neg ((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a)) \rightarrow \\
\mid (\neg \rightarrow) \\
\rightarrow ((\neg a \Rightarrow b) \Rightarrow (\neg b \Rightarrow a)) \\
\mid (\rightarrow \Rightarrow) \\
(\neg a \Rightarrow b) \rightarrow (\neg b \Rightarrow a) \\
\mid (\rightarrow \Rightarrow) \\
(\neg a \Rightarrow b), \neg b \rightarrow a \\
\mid (\neg \rightarrow) \\
(\neg a \Rightarrow b) \rightarrow b, a \\
\wedge (\Rightarrow \Rightarrow) \\
\rightarrow \neg a, b, a & \quad b \rightarrow b, a \\
\mid (\rightarrow \neg) \quad \text{axiom} \\
a \rightarrow b, a & \quad \text{axiom}
\end{align*}
\]
Solution
The tree $T \rightarrow A$ is not a proof in $GL$ because a rule corresponding to the decomposition step below does not exist in $GL$

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

$$| (\neg \rightarrow)$$

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

The tree $T \rightarrow A$ is a proof is some system $GL1$ that has all the rules of $GL$ except of its $(\neg \rightarrow)$ rule:

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

This $GL$ rule has to be replaced in $GL1$ by the rule:

$$(\neg \rightarrow)_1 \quad \frac{\Gamma, \Gamma' \rightarrow \Delta, A, \Delta'}{\Gamma, \neg A, \Gamma' \rightarrow \Delta, \Delta'}$$
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 prove the strong soundness property of the proof system GL.
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 $\Gamma \rightarrow \Delta \in SQ$, such that its decomposition tree $T_{\Gamma \rightarrow \Delta}$ 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 $\Gamma \rightarrow \Delta$. 
Counter Model

In particular, given a sequent

$$\rightarrow A$$

and its decomposition tree

$$T_{\rightarrow A}$$

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 $\Gamma \rightarrow \Delta$, such that its decomposition tree $T_{\Gamma \rightarrow \Delta}$ contains a non-axiom leaf $L_A$
Any truth assignment $v$ that falsifies the non-axiom leaf $L_A$ is a counter model for $\Gamma \rightarrow \Delta$

In particular, given a formula $A \in \mathcal{F}$, and its decomposition tree $T_A$ with a non-axiom leaf, this leaf let us define a counter-model for $A$ determined by the decomposition tree $T_A$
Exercise

We know that the system \textbf{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

\[ T_{\rightarrow A} \]

\[ \rightarrow ((b \rightarrow a) \rightarrow (\neg b \rightarrow a)) \]

\[ | (\rightarrow \Rightarrow) \]

\[ (b \Rightarrow a) \rightarrow (\neg b \Rightarrow a) \]

\[ | (\rightarrow \Rightarrow) \]

\[ \neg b, (b \Rightarrow a) \rightarrow a \]

\[ | (\neg \rightarrow) \]

\[ (b \Rightarrow a) \rightarrow b, a \]

\[ \wedge (\Rightarrow \Rightarrow) \]

\[ \rightarrow b, b, a \]

\[ a \rightarrow b, a \]

non – axiom

axiom
Exercise

The non-axiom leaf \( L_A \) we want to falsify is \[ \rightarrow b, b, a \]

Let \( v : \text{VAR} \rightarrow \{T, F\} \) be a truth assignment.

By definition of semantic for sequents we have that

\[ v^*(\rightarrow b, b, a) = (T \Rightarrow v(b) \cup v(b) \cup v(a)) \]

Hence \( v^*(\rightarrow b, b, a) = F \) 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 \( T_A \rightarrow \) is any \( v : \text{VAR} \rightarrow \{T, F\} \) such that

\[ v(b) = v(a) = F \]
Our goal now is to prove the Completeness Theorem for GL.

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

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

Moreover

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

$$\vdash_{GL} \Gamma \rightarrow \Delta \quad \text{if and only if} \quad \models \Gamma \rightarrow \Delta$$
GL Completeness

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

\[
\text{if } \models A \text{ then } \vdash_{\text{GL}} A
\]

We prove instead of the logically equivalent opposite implication:

\[
\text{if } \rhd_{\text{GL}} A \text{ then } \not\models A
\]
**GL Completeness**

Assume $\forall_{GL} A$, i.e. $\forall_{GL} \rightarrow A$

Let $\mathcal{T}_A$ be a set of all decomposition trees of $\rightarrow A$.

As $\forall_{GL} \rightarrow A$ each tree $T_A$ in the set $\mathcal{T}_A$ has a non-axiom leaf. We choose an arbitrary $T_A \in \mathcal{T}_A$.

Let $L_A = \Gamma' \rightarrow \Delta'$ be a non-axiom leaf of $T_A$.

We define a truth assignment $v : \text{VAR} \rightarrow \{T, F\}$ which falsifies $L_A = \Gamma' \rightarrow \Delta'$ as follows:

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

By Counter Model Theorem

$\not\models A$