cse541
LOGIC FOR COMPUTER SCIENCE

Professor Anita Wasilewska

Spring 2015

LECTURE 10

Chapter 10
CLASSICAL AUTOMATED PROOF SYSTEMS

PART 1: **RS** SYSTEM

PART 2: **RS1, RS2, RS3** SYSTEMS

PART 3: GENTZEN SYSTEMS

CLASSICAL AUTOMATED PROOF SYSTEMS

Hilbert style systems are easy to define and admit a relatively simple proofs of the Completeness Theorem but they are difficult to use

Automated systems are less intuitive then the Hilbert-style systems, but they will allow us to define effective automatic procedures for **proof search**, what is impossible in a case of the Hilbert-style systems

The first idea of this type was presented by **G. Gentzen** in 1934

We present in this chapter our version of original Gentzen system for **propositional classical logic**

We present the original Gentzen systems for **Intuitionistic** and **Classical** Propositional Logics in Chapter 12

AUTOMATED PROOF SYSTEMS

PART 1: RS System

The automated proof system we presented here is due to Helena Rasiowa and Roman Sikorski

We present here the propositional version of the original system and call it RS system for Rasiowa - Sikorski

The propositional RS system extends naturally to predicate logic QRS system which is presented in Chapter 14

Both systems RS and QRS admit a **constructive proof** of **Completeness Theorem**

First such constructive proofs were given, together with the formalization of the systems by H. Rasiowa and Sikorski in 1961

AUTOMATED PROOF SYSTEMS

PART 2: RS1, RS2, RS3 System

We define, as an exercise 3 versions of of the RS System, discuss their differences and show how the proof of **Completeness Theorem** for RS **extends** to similar proofs for all 3 systems

AUTOMATED PROOF SYSTEMS

PART 3: GENTZEN Systems - Lecture 13

We present our modern versions of Gentzen Sequent systems for propositional classical logic

Both systems **extend** easily to predicate logic and admit a **constructive proof** of **Completeness Theorem** via Rasiowa-Sikorski method

The original Gentzen system **LK** for classical propositional logic is presented in Chapter 12 together with the original Gentzen system **LI** for the Intuitionistic propositional logic

PART1:
RS Proof System for Classical Propositional Logic

RS Proof System

Language of **RS** is

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

The rules of inference of our system **RS** operate on **finite sequences** of formulas and we adopt

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

as the set of expressions of **RS**

Notation

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

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

with indices if necessary.

RS Proof System

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

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

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

we **denote**

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

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

Formal Semantics for RS

Let $v : VAR \longrightarrow \{T, F\}$ be a truth assignment and v^* its classical semantics extension to the set of formulas $\mathcal{F}$

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

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

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

We write it as

$$v \models \Gamma$$

and call v a **model** for Γ

Formal Semantics for RS

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

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

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

We write as always,

$$\models \Gamma$$

to denote that Γ is a **tautology**

Example

Example

Let Γ be a sequence

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

The truth assignment v such that

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

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

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

Rules of inference

Rules of inference of **RS** are of the form:

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

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

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

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

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

Rules of inference of RS

Proof System **RS** contains seven inference rules:

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

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

Definition

Any propositional variable, or a negation of propositional variable is called a **literal**

The set

$$LT = VAR \cup \{\neg a : \quad a \in VAR\}$$

is called a set of all propositional **literals**

The variables are called **positive literals**

Negations of variables are called **negative literals**.

Literal

We denote by

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

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

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

We will denote by

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

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

Logical Axioms of RS

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

$$\Gamma_1^{'},\ a,\ \Gamma_2^{'},\ \neg a,\ \Gamma_3^{'}$$

$$\Gamma_1^{'},\ \neg a,\ \Gamma_2^{'},\ a,\ \Gamma_3^{'}$$

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

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

Inference Rules of RS

Disjunction rules

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

Conjunction rules

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

Inference Rules of RS

Implication rules

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

Negation rule

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

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

Proof System RS

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

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

where the set of inference rules is

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

and LA is the set of all logical axioms, as defined before

Proof Trees

Definition

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

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

built out of sequences satisfying the following conditions:

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

2. all leafs are axioms

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

Proof Trees

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

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

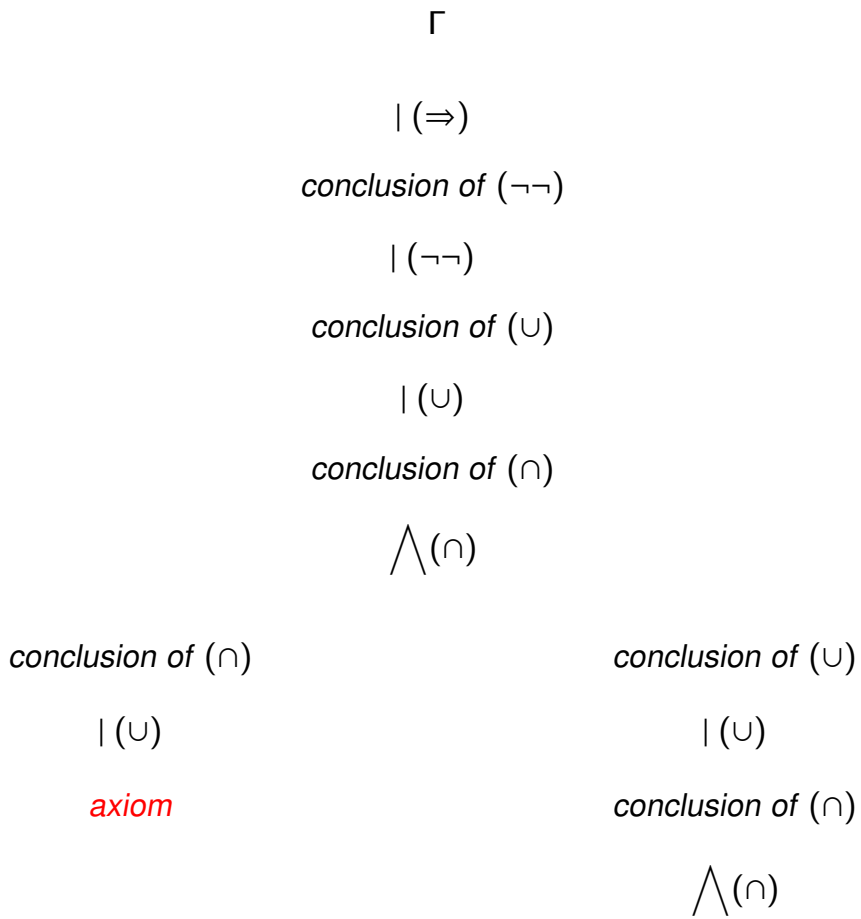
Example

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

We represent it as the following tree

Proof Trees

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



Proof Trees

The Proof Trees represent a certain visualization for the proofs

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

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

Example

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

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

is the as follows

Proof Trees

The tree $\mathbf{T}_A$

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

$$| \ (\Rightarrow)$$

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

$$| \ (\neg\neg)$$

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

$$\bigwedge (\cap)$$

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

$$| \ (\cup) \qquad\qquad | \ (\cup)$$

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

Formal Proof

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

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

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

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

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

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

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

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

Example

Example

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

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

consists of building a certain tree and proceeds as follows.

Example

The tree $\mathbf{T}_A$

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

$$| \ (\Rightarrow)$$

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

$$| \ (\neg\neg)$$

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

$$| \ (\cup)$$

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

$$\bigwedge (\cap)$$

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

Example

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

Here it is

$$a, b, \neg b$$

$$a, b, \neg a$$

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

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

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

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

Decomposition Trees

Our GOAL in inventing proof systems like **RS** is to facilitatee automatic proof search

The method of such proof search is to generate what is called the **decomposition trees**

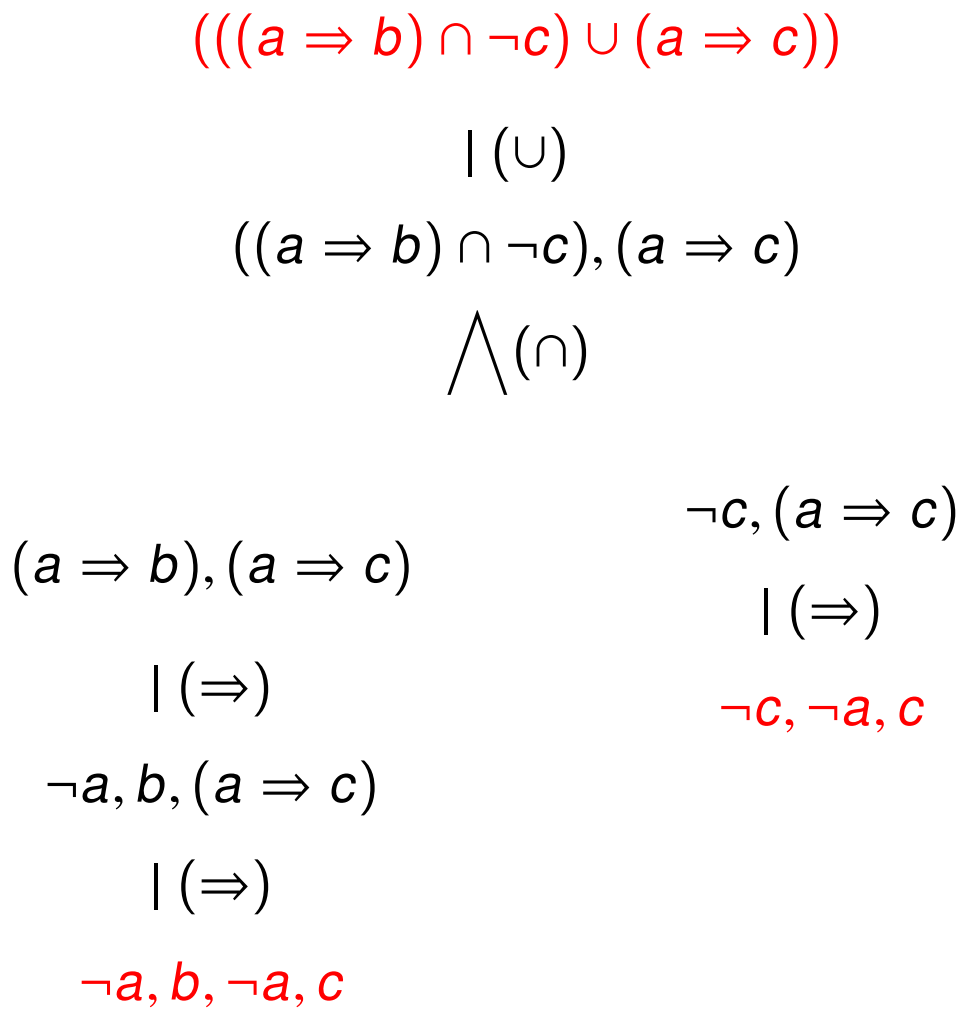
The decomposition tree for

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

is built as follows

Decomposition Trees

The tree $\mathbf{T}_A$



Decomposition Trees

Observe that the decomposition tree $\mathbf{T}_A$ contains an **non-axiom leaf**

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

hence **it is not a proof** of A in **RS**

Moreover, we are going to prove that a the decomposition trees in **RS** are always **unique**

From the **uniqueness** of $\mathbf{T}_A$ we have that if $\mathbf{T}_A$ has a **non-axiom leaf** then the **proof** of A in **RS** does not exist

This fact becomes crucial in our proof of **Completeness Theorem**

Counter Models

The other crucial idea used in the proof of **Completeness Theorem** is that of a Counter Model defined by a decomposition tree

Example Given a formula A

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

and its decomposition tree $\mathbf{T}_A$ presented on the next slide

Counter Models

The tree $\mathbf{T}_A$

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

$$| \ (\cup)$$

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

$$\bigwedge (\cap)$$

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

$$| \ (\Rightarrow)$$

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

$$| \ (\Rightarrow)$$

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

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

$$| \ (\Rightarrow)$$

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

Counter Models

Consider a **non-axiom leaf** of $\mathbf{T}_A$

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

Let now v be any variable assignment

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

such that it makes this non-axiom leaf **false**

We put constructing a v **restricted** to VAR_A

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

Obviously, we have that

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

Strong Soundness

Moreover, we are going the **prove** that all the rules of inference of **RS** of are **strongly sound,** i.e.

$$C \equiv P, \qquad C \equiv P_1 \cap P_2$$

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

We use the strong soundness of the rules to **prove**, by induction on the degree of sequences on a branch of $\mathbf{T}_A$ that starts with the formula A and ends with a non-axiom leaf, that any v that make this non-axiom leaf **false** also falsifies all sequences on the branch and hence falsifies the formula A

This means that $v \not\models A$, i.e. v is a **counter-moldel** for A

Counter Models

Consider a branch of of $\mathbf{T}_A$ with the **non-axiom leaf**

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

In particular, the formula A is on this branch, hence we get that

$$v^*(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c)) = F$$

and v is a **counter-model** for A

Definition

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

Completeness Theorem

The construction of the counter-models generated by the decomposition trees is another **crucial point** to the proof of the Completeness Theorem for **RS**.

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

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

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

and then we generalize it to the following

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

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

Strong Soundness of RS

Strong Soundness

Definition

Given a proof system

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

Definition

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

Definition

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

Strong Soundness of RS

Fact

The proof system **RS** is strongly sound

Proof

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

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

By definition of strong soundness we have to show that

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

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

in case of one premiss rule and

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

in case of the two premisses rule.

Strong Soundness of RS

Consider the rule $(\cup)$

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

We evaluate:

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

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

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

Strong Soundness of RS

Consider the rule $(\neg\cup)$

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

We evaluate:

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

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

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

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

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

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

Soundness Theorem

Observe that the strong soundness notion implies soundness (not only by name!) and obviously all LA of **RS** are **tautologies** hence we have also proved the following

Soundness Theorem for **RS**

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

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

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

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

Completeness Theorem

Our goal now it to prove **Completeness Part** of the **Completeness Theorem**, i.e. to prove that the following holds

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

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

We prove instead the opposite implication

RS Completeness Part

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

Here are main steps and facts needed for proof

Step 1 Define, for each $A \in \mathcal{F}$ its **decomposition tree** $\mathbf{T}_A$

Completeness Theorem

Step 2 Prove the following Lemmas

Lemma 1

For any $A \in \mathcal{F}$, the decomposition tree $\mathbf{T}_A$ is **unique**

Lemma 2

For any $A \in \mathcal{F}$, $\mathbf{T}_A$ has the following property:

$\nvdash_{\mathbf{RS}} A$ if and only if **there is a leaf** of $\mathbf{T}_A$ which is **not an axiom**

Completeness Theorem

Lemma 3

For any $A \in \mathcal{F}$, such that $\mathbf{T}_A$ has a **non-axiom** leaf, and for any truth assignment v, such that

$$v^*(\text{non-axiom leaf}) = F$$

the v also **falsifies** A , i.e.

$$v^*(A) = F$$

Proof of Completeness Theorem

Proof of **Completeness Theorem**

Assume that A is any formula is such that

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

By **Lemma 2** the decomposition tree $\mathbf{T}_A$ contains a non-axiom leaf

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

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

This proves **Lemma 3** that

$$\not\models A$$

RS: DECOMPOSITION TREES

Decomposition Trees

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

Building a **decomposition tree**, i.e. a proof search tree consists in the **first step** of transforming the **RS rules** into corresponding **decomposition rules**

RS Decomposition Rules

Here are all of **RS** **decomposition rules**

Disjunction decomposition rules

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

Conjunction decomposition rules

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

Decomposition Rules

Implication decomposition rules

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

Negation decomposition rule

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

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

Tree Decomposition Rules

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

Tree Decomposition Rules

$(\cup)$ **rule**

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

$$| \, (\cup)$$

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

Tree Decomposition Rules

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

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

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

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

$(\cap)$ **rule**

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

$$\bigwedge(\cap)$$

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

Tree Decomposition Rules

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

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

$$| \ (\neg\cap)$$

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

$(\Rightarrow)$ **rule**

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

$$| \ (\cup)$$

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

Tree Decomposition Rules

$(\neg \Rightarrow)$ rule

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

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

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

$(\neg\neg)$ rule

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

$$| \, (\neg\neg)$$

$$\Gamma', A, \Delta$$

Definitions and Observations

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

Definition: Indecomposable Sequence

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

Definitions and Observations

Definition: Decomposable Formula

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

Definition: Decomposable Sequence

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

Definitions and Observations

Observation 1

Decomposition rules are functions with disjoint domains, i.e.

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

there is **exactly one** decomposition rule that can be applied to it

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

Definitions and Observations

Observation 2

If the main connective of the **first** decomposable formula is $\cup, \cap, \Rightarrow,$

then the **decomposition rule** determined by it is $(\cup), (\cap), (\Rightarrow)$, respectively

Observation 3

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

then the **decomposition rule** is determined by the **second connective** of the formula A

The corresponding **decomposition rules** are $(\neg\cup), (\neg\cap), (\neg\neg), (\neg \Rightarrow)$

Lemma

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

Unique Decomposition Lemma

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

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

Decomposition Tree Definition

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

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

Step 1.

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

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

Step 2.

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

Decomposition Tree Definition

Step 3.

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

By the **Unique Decomposition Lemma** and **Observations 2,3** 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 steps 2 and 3 until we obtain only leaves

Decomposition Theorem

We now our **Lemmas 1,2 3** needed for the proof of the Completeness Theorem into one

Decomposition Tree Theorem

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

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

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

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

Theorem

Proof

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

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

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

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

Example

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

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

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

The first decomposition rule used is determined by its main connective

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

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

Example

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

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

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

$$| \ (\cup)$$

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

Example

Now we decompose the sequence

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

It is a **decomposable** sequence with the first, decomposable formula

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

The next step of the construction of our decomposition tree is determined by its main connective $\Rightarrow$ and we put the box around it

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

Example

The decomposition tree becomes now

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

$$|\ (\cup)$$

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

$$|\ (\Rightarrow)$$

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

Example

The next sequence to decompose is

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

with the first decomposable formula

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

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

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

Example

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

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

$$| \ (\cup)$$

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

$$| \ (\Rightarrow)$$

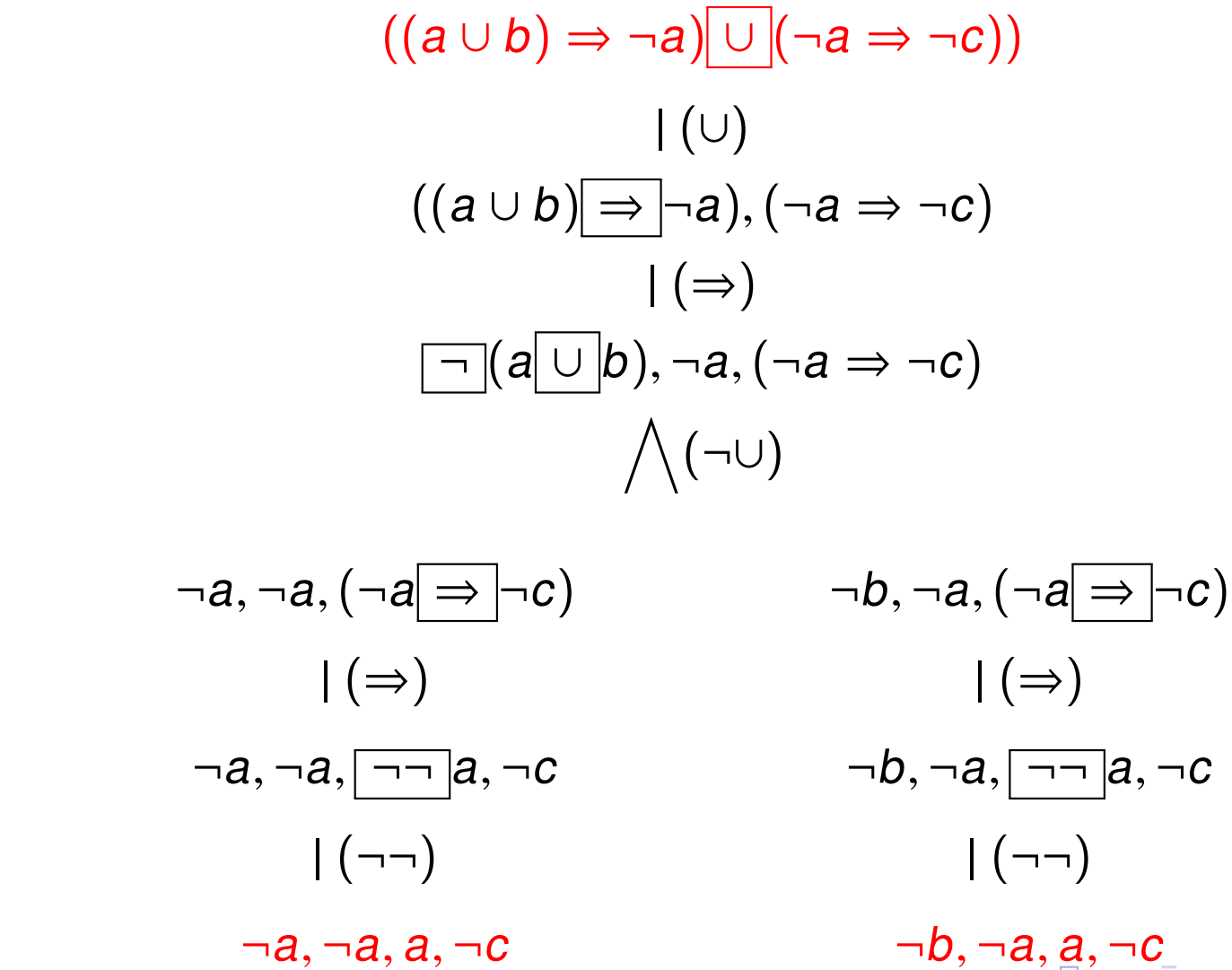
$$\boxed{\neg}(a \boxed{\cup} b), \neg a, (\neg a \Rightarrow \neg c)$$

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

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

Example

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



Example

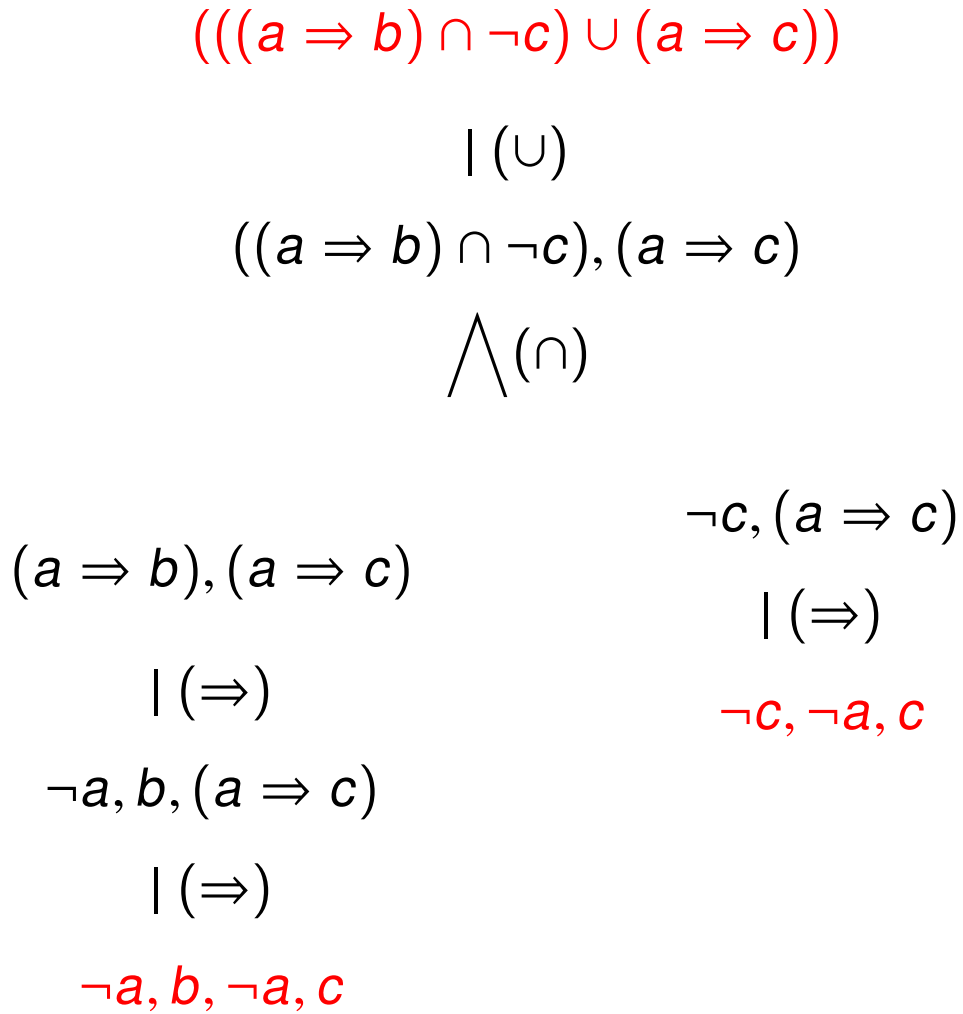
All leaves of $\mathbf{T}_A$ are axioms

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

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

Example

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



Counter Model

Consider a non-axiom leaf of $\mathbf{T}_A$

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

We will define now define a counter-model **generated** by a decomposition tree $\mathbf{T}_A$

v , by definition is any variable assignment

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

- that makes this non-axiom leaf **false** i.e. for example we put

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

Obviously, we have that

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

Counter Model

We have **proved** that **RS** is strongly sound

The strong soundness of the rules means that if **one of premisses** of a rule is **false**, so is the **conclusion**

Hence, the strong soundness of the rules **proves**, by induction on the degree of sequences $\Gamma \in \mathbf{T}_A$, which v that made a leaf false **falsifies** all sequences on the **branch** of $\mathbf{T}_A$ that ends with the already **falsified** leaf

Counter Model Theorem

We have hence proved the following

Counter Model Theorem

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

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

Counter Model

In particular, the formula A belongs to the branch with **falsified** non- axiom leaf

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

By the **Counter Model Theorem**

$$v^*(A) = v^*(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c)) = F$$

i.e. v is **a counter-model** for A and we proved that

$$\not\models A$$

Counter Model

F ”climbs” the tree $\mathbf{T}_A$

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

$$| \, (\cup)$$

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

$$\bigwedge (\cap)$$

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

$$| \, (\Rightarrow)$$

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

$$| \, (\Rightarrow)$$

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

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

$$| \, (\Rightarrow)$$

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

axiom

Counter Model

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

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

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

In our case the formula A has only one non-axiom leaf, and hence only one restricted **counter model**

Completeness Theorem Revisited

RS Completeness Theorem

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

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

We prove instead the opposite implication

RS Completeness Theorem

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

Proof of Completeness Theorem

Proof of **Completeness Theorem**

Assume that A is any formula is such that

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

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

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

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

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

$$\not\models A$$

PART2:
RS1, RS2, RS3 Proof Systems

RS1 Proof System

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

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

The rules of inference of our system **RS1** operate as rules of **RS** on **finite sequences** of formulas and we adopt

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

as the set of expressions of **RS1**

Notation

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

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

with indices if necessary.

Rules of inference of RS1

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

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

The inference rules of **RS1** are quite similar to the rules of **RS** - look at them CAREFULLY! to see where lies the difference!

REMINDER: **Definition**

Any propositional variable, or a negation of propositional variable is called a **literal**

The set

$$LT = VAR \cup \{\neg a : \quad a \in VAR\}$$

is called a set of all propositional **literals**

The variables are called **positive literals**

Negations of variables are called **negative literals**.

Literals Notation

We denote, as before, by

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

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

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

We will denote by

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

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

Logical Axioms of RS1

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

Logical Axioms LA of **RS1** are as follows

$$\Gamma_1', \ a, \ \Gamma_2', \ \neg a, \ \Gamma_3'$$

$$\Gamma_1', \ \neg a, \ \Gamma_2', \ a, \ \Gamma_3'$$

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

Inference Rules of RS1

Disjunction rules

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

Conjunction rules

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

Inference Rules of RS1

Implication rules

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

Negation rule

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

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

Proof System RS1

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

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

where

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

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

System RS1

Exercise

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

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

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

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

4. Prove **Completeness Theorem** for **RS1**

System RS1

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

$\mathbf{T}_A$

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

$$| \ (\Rightarrow)$$

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

$$| \ (\cup)$$

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

$$| \ (\neg\neg)$$

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

$$\bigwedge (\cap)$$

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

Strong Soundness of RS1

2. Observe that the system **RS1** is obtained from **RS** by changing the sequence Γ' into Γ and the sequence Δ into Δ' in **all** of the rules of inference of **RS**

These changes do not influence the essence of proof of **strong soundness** of the rules of **RS**

One has just to replace the sequence Γ' by Γ and Δ by Δ' in the the proof of **strong soundness** of each rule of **RS** to obtain the corresponding proof of **strong soundness** of corresponding rule of **RS1**

We do it, for example for the rule $(\cup)$ of **RS1** as follows

Strong Soundness of RS1

Consider the rule $(\cup)$ of **RS1**

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

We evaluate:

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

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

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

Decomposition Trees in RS1

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

We follow now the following steps

Step 1

Decompose using rule defined by the main connective of a decomposable formula B

Step 2

Traverse resulting sequence Γ on the new node of the tree from RIGHT to LEFT and **find** first decomposable formula

Step 3

Repeat **Step 1** and **Step 2** until no more decomposable formulas

End of Tree Construction

Decomposition Trees in RS1

4.

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

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

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

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

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

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

Proof of Completeness Theorem for RS1

Proof of **Completeness Theorem**

Assume that A is any formula is such that

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

By **Fact 2** the decomposition tree $\mathbf{T}_A$ contains a non-axiom leaf

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

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

This proves that

$$\not\models A$$

System RS2

Definition

System **RS2** is a proof system obtained from **RS** by changing the sequences Γ' into Γ in **all of the rules** of inference of **RS**

The logical axioms LA remind the same

Exercises

E1 Construct **two** decomposition trees in **RS2** of the formula

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

E2 Show that **RS2** is **strongly sound**

E3 Prove the **Soundness Theorem** for **RS2**

System RS2

Exercises

E3 Define shortly, in your own words, for any formula A, its decomposition tree $\mathbf{T}_A$ in **RS2**

Justify why your definition is correct

Show that in **RS2** the decomposition tree for as given formula A may not be unique

E4 Prove the **Completeness Theorem** for **RS2**

System RS2

Exercise

Write a procedure $TREE_A$ such that for any formula A of **RS2** it produces its UNIQUE decomposition tree and prove **COMPLETENESS** of this procedure

System RS3

Definition

System **RS2** is a proof system obtained from **RS** by changing its LA to the following set of axioms

The rules of inference remind the same

$$\Gamma_1,\ A,\ \Gamma_2,\ \neg A,\ \Gamma_3$$

$$\Gamma_1,\ \neg A,\ \Gamma_2,\ A,\ \Gamma_3$$

where $A \in \mathcal{F}$ is any **formula**

We denote by LA the set of all logical axioms of **RS3**

System RS3

Prove the **Completeness Theorem** for **RS3**