Chapter 11: Automated Proof Systems (1)

SYSTEM RS OVERVIEW

Hilbert style systems are easy to define and admit a simple proof 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 give an effective automatic procedure for proof search, what was impossible in a case of the Hilbert-style systems.

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

PART 1: RS SYSTEM

RS proof system presented here is due to **H. Rasiowa and R. Sikorski** and appeared for the first time in 1961. It extends easily to Predicate Logic and admits a CONSTRUCTIVE proof of Completeness Theorem (first given by Rasiowa- Sikorski).

PART 2: GENTZEN SYSTEM

We present two Gentzen Systems; a modern version and the original version. BOTH extend easily to Predicate Logic and admit a CONSTRUCTIVE proof of Completeness Theorem via Rasiowa-Sikorski method. The Original Gentzen system is easily adopted to a complete system foir the Intuitionistic Logic and will be presented in Chapter 12.

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*.

Set of expressions $\mathcal{E} = \mathcal{F}^*$.

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

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

with indices if necessary.

Meaning of Sequences: 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 define

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

Formal Semantics for RS Let $v : VAR \longrightarrow \{T, F\}$ be a truth assignment, 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.

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

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

Such a truth assignment v is called *a model* for Γ.

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

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

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

Example Let Γ be a sequence

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

The truth assignment v for which $v(a) = F$ and $v(b) = T$ 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 of **RS** are of the form:

$$\frac{\Gamma_1}{\Gamma} \quad \textit{or} \quad \frac{\Gamma_1 \; ; \; \Gamma_2}{\Gamma},$$

where Γ_1, Γ_2 and Γ are sequences Γ_1, Γ_2 are called premisses and Γ is called the conclusion of the rule of inference.

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.

System RS contains seven inference rules:

$$(\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg \Rightarrow), \text{ and } (\neg\neg).$$

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

Literals

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

The variables are called *positive literals*.

Negations of variables are called *negative literals*.

We denote by $\Gamma^{'}, \Delta^{'}, \Sigma^{'}$ finite sequences (empty included) formed out of *literals* i.e

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

We will denote by Γ, Δ, Σ the elements of $\mathcal{F}^{*}$.

Axioms $\mathcal{AL}$ of RS We adopt as an axiom any sequence which contains any propositional variable and its negation, i.e any sequence

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

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

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

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

Implication rules

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

Negation rule

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

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

The Proof System RS Formally we define:

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

Proof Tree By a proof tree, or **RS**-proof of Γ we understand a tree $\mathbf{T}_\Gamma$ of sequences satisfying the following conditions:

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

2. all *leafs* are axioms,

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

We picture, and write our proof trees with the node on the top, and leafs on the very bottom, instead of more common way, where the leafs are on the top and root is on the bottom of the tree.

We write our proof trees indicating additionally the name of the inference rule used at each step of the proof.

For example, if the proof of a *theorem* from *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:

$$theorem; \quad provable \quad formula$$

$$\mid (\Rightarrow)$$

$$conclusion\ of\ (\neg\neg)$$

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

$$conclusion\ of\ (\cup)$$

$$\mid (\cup)$$

$$conclusion\ of\ (\cap)$$

$$\bigwedge (\cap)$$

$$conclusion\ of\ (\cap) \quad conclusion\ of\ (\cup)$$

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

$$axiom \qquad conclusion\ of\ (\cap)$$

$$\bigwedge (\cap)$$

$$axiom \quad axiom$$

Trees represent a certain *visualization* for the proofs and any formal proof in any system can be represented in a tree form.

Example The proof tree in **RS** of the de Morgan law

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

is the following.

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

$$\mid (\Rightarrow)$$

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

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

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

$$\bigwedge(\cap)$$

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

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

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

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

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

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

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

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

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

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

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

The search for the proof of $(\neg(a \cup b) \Rightarrow (\neg a \cap \neg b))$ consists of building a certain tree and proceeds as follows.

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

$$\mid (\Rightarrow)$$

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

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

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

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

We construct its formal proof, written in a vertical manner, by writing the two axioms, which form the two premisses of the rule $(\cap)$ one above the other. All other sequences remain the same.

$$a, b, \neg b$$

$$a, b, \neg a$$

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

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

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

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

The tree generated by the proof search is called *a decomposition tree*.

Example of decomposition tree for $(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$ is the following.

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

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

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

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

$$\mid (\Rightarrow)$$

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

The decomposition tree generated by this search contains an non-axiom leaf and hence is not a proof.

Moreover, it proves, as the decomposition (proof search) tree is unique that the proof of the formula

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

DOES not EXIST in the system **RS**.

Counter-model generated by the decomposition tree.

Example: Given a formula A:

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

and its decomposition tree $\mathbf{T}_A$.

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

$$\mid (\cup)$$

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

$$\bigwedge (\cap)$$

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

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

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

$$\mid (\Rightarrow)$$

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

Consider a non-axiom leaf:

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

Let v be any variable assignment

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

such that it makes this non-axiom leaf False, i.e. we put

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

Obviously, we have that

$$v^{*}(\neg a, b, \neg a, c) = F.$$

Moreover, all our rules of inference are sound (to be proven formally in the next section).

Rules soundness means that if one of premisses of a rule is FALSE, so is the conclusion.

Hence, the soundness of the rules proves (by induction on the degree of sequences $\Gamma \in \mathbf{T}_A$) that v, as defined above falsifies all sequences on the branch of $\mathbf{T}_A$ that ends with the non-axiom leaf $\neg a, b, \neg a, c$.

In particular, the formula A is on this branch, hence

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

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

The truth assignments defined by a non- axiom leaves are called **counter-models generated** by the decomposition tree.

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

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

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

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

and then we deduce from it the following full Completeness Theorem for sequences $\Gamma \in \mathcal{F}^*$.

Completeness Theorem 2

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

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

The Completeness Theorem consists of two parts:

Soundness Part: (Soundness Theorem) for any $A \in \mathcal{F}$,

if $\vdash_{\mathbf{RS}} A$, then $\models A$.

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

if $\models A$, then $\vdash_{\mathbf{RS}} A$.

Soundness Theorem for RS

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

if $\vdash_{\mathbf{RS}} \Gamma$, then $\models \Gamma$.

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

if $\vdash_{\mathbf{RS}} A$, then $\models A$.

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

We show even more, that the premisses and conclusion of both rules are logically equivalent.

If $P_1, (P_2)$ are premiss(es) of a rule, C is its conclusion, then

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

in case of one premiss rule and

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

in case of the two premisses rule.

Consider the rule $(\cup)$.

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

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

Consider the rule $(\neg\cup)$.

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

We evaluate: $v^*(\Gamma^{'}, \neg A, \Delta) \cap v^*(\Gamma^{'}, \neg B, \Delta) = (v^*(\Gamma^{'}) \cup v^*(\neg A) \cup v^*(\Delta)) \cap (v^*(\Gamma^{'}) \cup v^*(\neg B) \cup v^*(\Delta)) = (v^*(\Gamma^{'}, \Delta) \cup v^*(\neg A)) \cap (v^*(\Gamma^{'}, \Delta) \cup v^*(\neg B)) =$ by distributivity $= (v^*(\Gamma^{'}, \Delta) \cup (v^*(\neg A) \cap v^*(\neg B)) = v^*(\Gamma^{'}) \cup v^*(\Delta) \cup (v^*(\neg A \cap \neg B)) =$ by the logical equivalence of $(\neg A \cap \neg B)$ and $\neg(A \cup B)$ $= v^*(\delta_{\{\Gamma', \neg(A\cup B), \Delta\}} = v^*(\Gamma^{'}, \neg(A \cup B), \Delta))$.

We prove now the Completeness Part of the Completeness Theorem:

If $\not\vdash_{\mathbf{RS}} A$, then $\not\models A$.

STEPS needed for proof:

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

Step 2 (Lemma 1) Prove that the decomposition tree $\mathbf{T}_A$ is *unique*.

Step 3 (Lemma 2) Prove that $\mathbf{T}_A$ has the following property:

Proof of A in **RS** does not exist ($\not\vdash_{\mathbf{RS}}$)A, iff there is a leaf of $\mathbf{T}_A$ which is not an axiom.

Step 4 (Lemma 3) Prove that given A with $\mathbf{T}_A$ with a non-axiom leaf, we have that for any truth assignment v, such that v^*(non-axiom leaf) $= F$, v also falsifies A, i.e.

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

Proof of Completeness: Assume that A is any formula is such that

$$\not\vdash_{\mathbf{RS}} A.$$

By the STEP 3, the decomposition tree $\mathbf{T}_A$ contains a non-axiom leaf.

The non-axiom leaf $\mathcal{L}_A$ *defines* a truth assignment v which falsifies A, as follows:

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

This proves by STEP 4 that $\not\models A$.