
# Chapter 10

# Predicate Automated Proof Systems Completeness of Classical Predicate Logic

We define and discuss here a Rasiowa and Sikorski Gentzen style proof system **QRS** for classical predicate logic. The propositional version of it, the **RS** proof system, was studied in detail in Chap. 6. These both proof systems admit a *constructive proof* of completeness theorem. We adopt Rasiowa, Sikorski (1961) technique of construction a counter model determined by a decomposition tree to prove **QRS** completeness Theorem 10.4. The proof, presented in Sect. 10.3, is a generalization of the completeness proofs of **RS** and other Gentzen style propositional systems presented in details in Chap. 6. We refer the reader to this chapter as it provides a good introduction to the subject.

The other Gentzen type predicate proof system, including the original Gentzen proof systems **LK, LI** for classical and intuitionistic predicate logics are obtained from their propositional versions discussed in detail in Chap. 6. It can be done in a similar way as a generalization of the propositional **RS** the predicate **QRS** system presented here. We leave these generalizations as an exercises for the reader. That includes also the predicate language version of Gentzen proof of cut elimination theorem, Hauptzatz (1935). The Hauptzatz proof for the predicate classical **LK** and intuitionistic **LI** systems is easily obtained from the propositional proof included in Chap. 6.

There are of course other types of automated proof systems based on different methods of deduction.

The original version of this chapter was revised. A correction to this chapter is available at https://doi.org/10.1007/978-3-319-92591-2_12

**Electronic supplementary material** The online version of this chapter (https://doi.org/10.1007/978-3-319-92591-2_10) contains supplementary material, which is available to authorized users.



A. Wasilewska, *Logics for Computer Science*,
https://doi.org/10.1007/978-3-319-92591-2_10

There is a *Natural Deduction* mentioned by Gentzen in his Hauptzatz paper in 1935 and later fully developed by Dag Prawitz (1965). It is now called Prawitz, or Gentzen-Prawitz Natural Deduction.
There is a *Semantic Tableaux* deduction method invented by Evert Beth (1955). It was consequently simplified and further developed by Raymond Smullyan (1968). It is now often called Smullyan *Semantic Tableaux*.
Finally, there is a *Resolution*. The resolution method can be traced back to Davis and Putnam (1960). Their work is still known as Davis-Putnam method.The difficulties of their method were eliminated by John Alan Robinson (1965) and developed into what we call now Robinson Resolution, or just a Resolution.
There are many excellent textbooks covering each of these methods. We recommend Melvin Fitting book *First-order logic and automated theorem proving*(2nd ed.). Springer-Verlag(1996) as the one that not only covers all of them but also discusses their relationships.

The Resolution proof system for propositional or predicate logic operates on a set of *clauses* as a basic expressions and uses a *resolution rule* as the only rule of inference. In Sect. 10.4 we define and prove correctness of effective *procedures* of converting any formula $A$ into a corresponding set of clauses in both propositional and predicate cases. The correctness of propositional case is established by Theorem 10.5, of predicate case by Theorem 10.6. In the first step of the predicate procedure we define a process of *elimination of quantifiers* from the original language. It is called Skolemization of the language and is presented in Sect. 10.4.1. The correctness of the Skolemization is established by Skolem Theorem 10.11. In the second step of the procedure we show how convert a quantifiers free formula into logically equivalent set of clauses. It is presented with a proof of correctness (Theorem 10.13) in Sect. 10.4.2.

## 10.1 QRS Proof System

We define components and semantics of the proof system **QRS** as follows.
**Language $\mathcal{L}$**

We adopt a predicate (first order) language

$$\mathcal{L} = \mathcal{L}_{\{\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P},\mathbf{F},\mathbf{C}) \tag{10.1}$$

for **P, F, C** countably infinite sets of predicate, functional, and constant symbols respectively.
Let $\mathcal{F}$ denote a set of formulas of $\mathcal{L}$. The rules of inference of our system **QRS** operate on *finite sequences of formulas*, i.e. elements of $\mathcal{F}^*$ so we define the set of expressions of of **QRS** as follows. **Expressions $\mathcal{E}$**

We adopt as the set of expressions $\mathcal{E}$ of **RS** the set $\mathcal{F}^*$, i.e.

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

We will denote the expressions of **QRS**, i.e. the finite sequences of formulas by

$$\Gamma, \Delta, \Sigma, \text{with indices if necessary.}$$

In order to define the axioms $LA$ and the set of rules of inference of **QRS** we need to bring back some notions and to introduce some definitions.

An **atomic formula** of the predicate language $\mathcal{L}$ defined by (10.1) is any element of $\mathcal{A}^*$ (finite strings over the alphabet of $\mathcal{L}$) of the form

$$R(t_1, t_2, \ldots, t_n)$$

where $R \in \mathbf{P}, \#R = n$ and $t_1, t_2, \ldots, t_n \in \mathbf{T}$.

The set of all **atomic formulas** is denoted by $A\mathcal{F}$ and is defined as

$$A\mathcal{F} = \{R(t_1, t_2, \ldots, t_n) \in \mathcal{A}^* : \quad R \in \mathbf{P},\ t_1, t_2, \ldots, t_n \in \mathbf{T},\ n \geq 1\}. \tag{10.2}$$

We use symbols $R, Q, P, \ldots$ with indices if necessary to denote the **atomic formulas**.

**Literals**
We form a special subset $LT \subseteq \mathcal{F}$ of formulas, called a set of all *literals*, which is defined as follows.

$$LT = \{R : \ R \in A\mathcal{F}\} \cup \{\neg R : \ R \in A\mathcal{F}\}. \tag{10.3}$$

The atomic formulas (10.2) are called **positive literals** and the elements of the second set of the above union (10.3), i.e. the negations of the atomic formulas are called **negative literals**.

**Indecomposable, Decomposable Formulas**
A formula $A \in \mathcal{F}$ is *indecomposable* if and only if it is atomic or a negation of an atomic formula, i.e. an *literal*. Otherwise $A$ is *decomposable*.

Now we form *finite sequences* out of formulas (and, as a special case, out of literals). We need to distinguish the sequences formed out of literals from the sequences formed out of other formulas, so we adopt the following definition and notations.

**Indecomposable, Decomposable Sequences**
A sequence $\Gamma$ is *indecomposable* if and only if is formed out of indecomposable formulas only. Otherwise is *decomposable*.
We denote **indecomposable sequences** by by

$$\Gamma', \ \Delta', \ \Sigma', \ldots \quad \text{with indices if necessary.} \tag{10.4}$$

By definition, $\Gamma', \ \Delta', \ \Sigma' \ldots$ are finite sequences (empty included) formed out of **literals**, i.e $\Gamma', \ \Delta', \ \Sigma'\Gamma', \ \Delta', \ \Sigma' \in LT^*$.

We denote by

$$\Gamma, \ \Delta, \ \Sigma, \ldots \quad \text{with indices if necessary,} \tag{10.5}$$

the elements of $\mathcal{F}^*$, i.e. we denote $\Gamma,\ \Delta,\ \Sigma$ finite sequences (empty included) formed out of elements of $\mathcal{F}$.

**Logical Axioms LA**

As the *logical axiom* of **QRS** we adopt any sequence of formulas which contains a and its negation, i.e any sequence of the form

$$\Gamma_1,\ A,\ \Gamma_2,\ \neg A'\ \Gamma_3 \quad \text{or} \quad \Gamma_1,\ \neg A,\ \Gamma_2,\ A,\ \Gamma_3 \tag{10.6}$$

for any literal $A \in LT$ and any sequences $\Gamma_1, \Gamma_2, \Gamma_3 \in \mathcal{F}^*$ of formulas.

**Rules of inference** $\mathcal{R}$ (10.7)

**Group 1: Propositional Rules**

**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**

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

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

**Group 2: Quantifiers Rules**

$(\exists)$

$$\frac{\Gamma', A(t), \Delta, \exists x A(x)}{\Gamma', \exists x A(x), \Delta}$$

where $t$ is an arbitrary term.

$(\forall)$

$$\frac{\Gamma', A(y), \Delta}{\Gamma', \forall x A(x), \Delta}$$

where $y$ is a free individual variable which does not appear in any formula in the conclusion, i.e. in the sequence $\Gamma', \forall x A(x), \Delta$.

$(\neg\forall)$

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

$(\neg\exists)$

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

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

Note that $A(t), A(y)$ denotes a formula obtained from $A(x)$ by writing $t, y$, respectively, in place of all occurrences of $x$ in $A$. The variable $y$ in $(\forall)$ is called the *eigenvariable*. The condition: *where $y$ is a free individual variable which does not appear in any formula in the conclusion* is called the *eigenvariable condition*.

All occurrences of $y$ in $A(y)$ of the rule $(\forall)$ are fully indicated.

**The Proof System QRS**

Formally we define the proof system **QRS** as follows.

$$\mathbf{QRS} = (\mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}},\ \mathcal{E},\quad LA,\quad \mathcal{R}), \tag{10.8}$$

where $\mathcal{E} = \{\Gamma :\quad \Gamma \in \mathcal{F}^*\}$, $LA$ contains logical axioms of the system defined by (10.6), $\mathcal{R}$ is the set of rules of inference:

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

defined by (10.7).

By a **formal proof** of a sequence $\Gamma$ in the proof system **RS** we understand any sequence

$$\Gamma_1,\ \Gamma_2, \ldots . \Gamma_n \tag{10.9}$$

of sequences of formulas (elements of $\mathcal{F}^*$), such that

1. $\Gamma_1 \in LA,\ \Gamma_n = \Gamma$, and

2. for all i $(1 \leq i \leq n)$ $\Gamma_i \in AL$, or $\Gamma_i$ is a conclusion of one of the inference rules of **QRS** with all its premisses placed in the sequence $\Gamma_1,\ \Gamma_2,\ \ldots . \Gamma_{i-1}$.

We write, as usual,

$$\vdash_{QRS} \Gamma$$

to denote that $\Gamma$ has a formal proof in **QRS**.

As the proofs in **QRS** are sequences (definition of the formal proof) of sequences of formulas (definition of expressions $\mathcal{E}$) we will not use "," to separate the steps of the proof, and write the formal proof as $\Gamma_1;\ \Gamma_2;\ \ldots . \Gamma_n$.

We write, however, the formal proofs in **QRS** as we did the propositional case (Chap. 6), in a form of trees rather then in a form of sequences, ie. in a form of a tree, where *leafs* of the tree are axioms, *nodes* are sequences such that each sequence on the tree follows from the ones immediately preceding it by one of the rules. The *root* is a theorem. We picture, and write our tree-proofs 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 adopt hence the following definition.

**Definition 10.1 (Proof Tree)**

*By a proof tree, or* **QRS***-proof of* $\Gamma$ *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* $\Gamma$,

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 of inference (10.7).*

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.

In particular cases, as in the propositional case, we will write our proof- trees indicating additionally the name of the inference rule used at each step of the proof. For example, if in a proof of a formula $A$ from axioms (10.6) we use subsequently the rules

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

we represent the proof as the following tree denoted by $\mathbf{T}_A$.

$$\mathbf{T}_A$$

$$Formula\ A$$

$$|\ (\Rightarrow)$$

$$conclusion\ of\ (\forall)$$

$$|\ (\forall)$$

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

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

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

$$\bigwedge(\cap)$$

$$\begin{array}{cc} \textit{conclusion of } (\forall) & \textit{conclusion of } (\cup) \\ | \, (\forall) & | \, (\cup) \\ \textit{axiom} & \textit{conclusion of } (\cap) \\ & \bigwedge(\cap) \\ & \textit{axiom} \quad \textit{axiom} \end{array}$$

Remark that the derivation trees don't represent a different *definition* of a formal proof. This remains the same in the Gentzen – style systems. Trees represent a certain *visualization* for those proofs and any formal proof in any system can be represented in a tree form.

## 10.2 QRS Decomposition Trees

The main advantage of the Gentzen proof systems lies not in a way we generate proofs in them, but in the way we can *search* for proofs in them. That such proof searches happens to be deterministic and automatic. We conduct such search by treating inference rules as *decomposition rules* (see Chap. 6) and building *decomposition trees.*A general principle of building a decomposition tree is the following.

**Decomposition Tree $\mathbf{T}_\Gamma$**
For each $\Gamma \in \mathcal{F}^*$, a decomposition tree $\mathbf{T}_\Gamma$ is a tree build as follows.
**Step 1.** The sequence $\Gamma$ is the **root** of $\mathbf{T}_\Gamma$ and for any node $\Delta$ of the tree we follow the steps bellow.
**Step 2.** If $\Delta$ is indecomposable or an axiom, then $\Delta$ becomes a **leaf** of the tree.
**Step 3.** If $\Delta$ is decomposable, then we traverse $\Delta$ from left to right to identify the first **decomposable formula** $B$ and identify inference rule treated as decomposition rule determined uniquely by $B$. We put its left and right premisses as the left and right leaves, respectively.
**Step 4.** We repeat steps 2 and 3 until we obtain only leaves or infinite branch.

In particular case when $\Gamma$ has only one element, namely a formula $A \in \mathcal{F}$, we define we call it a decomposition tree of $A$ and denote by $\mathcal{T}_A$.

Here is a detailed definition of the decomposition tree for **QRS**.

**QRS Decomposition Tree Definition** (10.10)

Given a formula $A \in \mathcal{F}$, we define its decomposition tree $\mathbf{T}_A$ as follows.

Observe that the inference rules of **QRS** are divided in two groups: propo-

sitional connectives rules and quantifiers rules. The propositional connectives rules are: $(\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg \Rightarrow)$, and $(\neg\neg)$. The quantifiers rules are: $(\forall), (\exists), (\neg\forall)$ and $(\neg\exists)$.

We define the decomposition tree in the case of the propositional rules and the rules $(\neg\forall)$, $(\neg\exists)$ in the exactly the same way as in the propositional case (Chap. 6).

The case of the rules $(\forall)$ and $(\exists)$ is more complicated, as the rules contain the specific conditions under which they are applicable.

To define the way of decomposing the sequences of the form $\Gamma', \forall x A(x), \Delta$ or $\Gamma', \exists x A(x), \Delta$, i.e. to deal with the rules $(\forall)$ and $(\exists)$ in a way which would preserve the property of the *uniqueness* of the decomposition tree, we assume that all terms form a one-to one sequence

$$t_1, t_2, \ldots, t_n, \ldots\ldots \tag{10.11}$$

Observe, that by the definition, all free variables are terms, hence all free variables appear in the sequence (10.11). Let $\Gamma$ be a sequence on the tree with $\forall$ as a main connective, i.e. $\Gamma$ is of the form $\Gamma', \forall x A(x), \Delta$. We write a sequence $\Gamma', A(x), \Delta$ below it on the tree, as its child, where the variable $x$ has to fulfill the following condition.

**Condition 10.1** $(\forall)$

*$x$ is the first free variable in the sequence (10.11) such that $x$ does not appear in any formula in $\Gamma', \forall x A(x), \Delta$.*

Observe, that the condition 10.1 corresponds to the restriction put on the application of the rule $(\forall)$.

If the main connective of $\Gamma$, i.e. the main connective of the first formula in $\Gamma$ which is not an literal, is $(\exists)$. In this case $\Gamma$ is of the form $\Gamma', \exists x A(x), \Delta$, we write a sequence $\Gamma', A(t), \Delta, \exists x A(x)$ as its child, where the term $t$ has to fulfill the following condition.

**Condition 10.2** $(\exists)$

*$t$ is the first term in the sequence (10.11) such that the formula $A(t)$ does not appear in any sequence which is placed above $\Gamma', A(t), \Delta, \exists x A(x)$ on the tree.*

The fact that the sequence (10.11) is one- to- one and the fact that, by the conditions 10.1 and 10.2, we always chose the first appropriate term (variable) from this sequence, guarantee that the decomposition process is also unique in the case of the quantifiers rules $(\forall)$ and $(\exists)$.

From all above, and we conclude the following.

**Theorem 10.1**

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

**(i)** *the decomposition tree $\mathbf{T}_A$ is unique.*

**(ii)** *Moreover, the following conditions hold.*

*1. If $\mathbf{T}_A$ is* **finite** *and all its leaves are axioms, then*

$$\vdash_{QRS} A$$

*and $\mathbf{T}_A$ is a tree-proof of $A$ in* **QRS**.

*2. If $\mathbf{T}_A$ is* **finite** *and contains a non-axiom leaf, or $\mathbf{T}_A$ is* **infinite**, *then*

$$\nvdash_{QRS} A.$$

### 10.2.1 Examples of Decomposition Trees

In all the examples below, the formulas $A(x), B(x)$ represent any formula. But there is no indication about their particular components, so they are treated as **indecomposable formulas**.

**Example 10.1**

*The decomposition tree $\mathcal{T}_A$ of the de Morgan Law*

$$(\neg\forall x A(x) \Rightarrow \exists x \neg A(x))$$

*is the following.*

$$\mathbf{T}_A$$

$$(\neg\forall x A(x) \Rightarrow \exists x \neg A(x))$$

$$| \ (\Rightarrow)$$

$$\neg\neg\forall x A(x), \exists x \neg A(x)$$

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

$$\forall x A(x), \exists x \neg A(x)$$

$$| \ (\forall)$$

$$A(x_1), \exists x \neg A(x)$$

*where $x_1$ is a first free variable in the sequence (10.11) such that $x_1$ does not appear in $\forall x A(x), \exists x \neg A(x)$*

$$| \ (\exists)$$

$$A(x_1), \neg A(x_1), \exists x \neg A(x)$$

*where $x_1$ is the first term (variables are terms) in the sequence (10.11) such that $\neg A(x_1)$ does not appear on a tree above $A(x_1), \neg A(x_1), \exists x \neg A(x)$*

*Axiom*

The above tree $\mathbf{T}_A$ ended with an axiom, so it represents a proof of

$$(\neg \forall x A(x) \Rightarrow \exists x \neg A(x))$$

in **QRS**, i.e. we proved that

$$\vdash_{QRS} (\neg \forall x A(x) \Rightarrow \exists x \neg A(x)).$$

**Example 10.2**

*The decomposition tree $\mathbf{T}_A$ of*

$$(\forall x A(x) \Rightarrow \exists x A(x))$$

*is the following.*

$$\mathbf{T}_A$$

$$(\forall x A(x) \Rightarrow \exists x A(x))$$

$$| \ (\Rightarrow)$$

$$\neg \forall x A(x), \exists x A(x)$$

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

$$\neg \forall x A(x), \exists x A(x)$$

$$\exists x \neg A(x), \exists x A(x)$$

$$| \ (\exists)$$

$$\neg A(t_1), \exists x A(x), \exists x \neg A(x)$$

*where $t_1$ is the first term in the sequence (10.11), such that $\neg A(t_1)$ does not appear on the tree above $\neg A(t_1), \exists x A(x), \exists x \neg A(x)$*

$$| \ (\exists)$$

$$\neg A(t_1), A(t_1), \exists x \neg A(x), \exists x A(x)$$

*where $t_1$ is the first term in the sequence (10.11), such that $A(t_1)$ does not appear on the tree above $\neg A(t_1), A(t_1), \exists x \neg A(x), \exists x A(x)$*

*Axiom*

The above tree also ended with the axiom, hence we proved that

$$\vdash_{(} QRS) \ (\forall x A(x) \Rightarrow \exists x A(x)).$$

**Example 10.3**

*The decomposition tree* $\mathbf{T}_A$ *of*

$$(\exists x A(x) \Rightarrow \forall x A(x))$$

*is the following.*

$$\mathbf{T}_A$$

$$(\exists x A(x) \Rightarrow \forall x A(x))$$

$$| \ (\Rightarrow)$$

$$\neg \exists x A(x), \forall x A(x)$$

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

$$\forall x \neg A(x), \forall x A(x)$$

$$| \ (\forall)$$

$$\neg A(x_1), \forall x A(x)$$

*where* $x_1$ *is a first free variable in* (10.11) *such that* $x_1$ *does not appear in* $\forall x \neg A(x), \forall x A(x)$

$$| \ (\forall)$$

$$\neg A(x_1), A(x_2)$$

*where* $x_2$ *is a first free variable in* (10.11) *such that* $x_2$ *does not appear in* $\neg A(x_1), \forall x A(x)$*, the sequence* (10.11) *is one-to- one, hence* $x_1 \neq x_2$

*Non – axiom*

The decomposition tree, for any formula $A$ is *unique*, so we conclude from the fact that the above tree has a non-axiom branch that

$$\nvdash_{QRS} \ (\exists x A(x) \Rightarrow \forall x A(x)).$$

**Remark** when constructing the following tree $\mathbf{T}_A$ for the formula $\exists x A(x)$ in Example 10.4 below we adopt on the right branch of the a tree in the the shorthand notation instead of the repeating a similar reasoning performed on the left branch.

**Example 10.4**

*The decomposition tree* $\mathbf{T}_A$ *of the formula* $\exists x A(x)$ *is the following.*

$$\mathbf{T}_A$$

$$\exists x A(x)$$

$$| \ (\exists)$$

$$A(t_1), \exists x A(x)$$

*where* $t_1$ *is the first term in the sequence* (10.11)*, such that* $A(t_1)$ *does not appear on the tree above* $A(t_1), \exists x A(x)$

$$| \ (\exists)$$

$$A(t_1), A(t_2), \exists x A(x)$$

*where* $t_2$ *is the first term in the sequence* (10.11)*, such that* $A(t_2)$ *does not appear on the tree above* $A(t_1), A(t_2), \exists x A(x)$*, i.e.* $t_2 \neq t_1$

$$| \ (\exists)$$

$$A(t_1), A(t_2), A(t_3), \exists x A(x)$$

*where* $t_3$ *is the first term in the sequence* (10.11)*, such that* $A(t_3)$ *does not appear on the tree above* $A(t_1), A(t_2), A(t_3), \exists x A(x)$*, i.e.* $t_3 \neq t_2 \neq t_1$

$$| \ (\exists)$$

$$A(t_1), A(t_2), A(t_3), A(t_4), \exists x A(x)$$

$$| \ (\exists)$$

$$.....$$

$$| \ (\exists)$$

$$.....$$

$$infinite \ branch$$

Obviously, the above decomposition tree is infinite, what proves that

$$\nvdash_{QRS} \exists x A(x).$$

We will find now the proof of the distributivity law

$$(\exists x(A(x) \cap B(x)) \Rightarrow (\exists x A(x) \cap \exists x B(x)))$$

and show that we can't prove in **QRS** the inverse implication

$$((\exists x A(x) \cap \exists x B(x)) \Rightarrow \exists x(A(x) \cap B(x))).$$

**Remark** when constructing the following trees $\mathbf{T}_A$ in Examples 10.5 and 10.6 adopt, as we did in the previous Example 10.4, the shorthand notation when the reasoning is similar to the one presented in the Example 10.4.

**Example 10.5**

*The decomposition tree* $_A$ *of the first formula*

$$(\exists x(A(x) \cap B(x)) \Rightarrow (\exists xA(x) \cap \exists xB(x)))$$

*is the following.*

$$\mathbf{T}_A$$

$$(\exists x(A(x) \cap B(x)) \Rightarrow (\exists xA(x) \cap \exists xB(x)))$$

$$| \ (\Rightarrow)$$

$$\neg\exists x(A(x) \cap B(x)), (\exists xA(x) \cap \exists xB(x))$$

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

$$\forall x\neg(A(x) \cap B(x)), (\exists xA(x) \cap \exists xB(x))$$

$$| \ (\forall)$$

$$\neg(A(x_1) \cap B(x_1)), (\exists xA(x) \cap \exists xB(x))$$

*where* $x_1$ *is a first free variable in the sequence* (10.11) *such that* $x_1$ *does not appear in* $\forall x\neg(A(x) \cap B(x)), (\exists xA(x) \cap \exists xB(x))$

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

$$\neg A(x_1), \neg B(x_1), (\exists xA(x) \cap \exists xB(x))$$

$$\bigwedge(\cap)$$

$$\neg A(x_1), \neg B(x_1), \exists xA(x)$$

$$| \ (\exists)$$

$$\neg A(x_1), \neg B(x_1), A(t_1), \exists xA(x)$$

*where* $t_1$ *is the first term in the sequence* (10.11)*, such that* $A(t_1)$ *does not appear on the tree above* $\neg A(x_1), \neg B(x_1), A(t_1), \exists xA(x)$ *Observe, that it is possible that* $t_1 = x_1$*, as* $A(x_1)$ *does not appear on the tree above. By the definition of the sequence* (10.11)*,* $x_1$ *is placed somewhere in it, i.e.* $x_1 = t_i$*, for certain* $i \geq 1$*. It means that after* $i$ *applications of the step* $(\exists)$ *in the decomposition tree, we will get a step:*

$$| \ (\exists)$$

$$\neg A(x_1), \neg B(x_1), \ldots A(x_1), \exists xA(x)$$

$$\neg A(x_1), \neg B(x_1), \exists xB(x)$$

$$| \ (\exists)$$

$$\neg A(x_1), \neg B(x_1), B(t_1), \exists xB(x)$$

$$| \ (\exists)$$

$$\ldots$$

$$| \ (\exists)$$

$$\neg A(x_1), \neg B(x_1), \ldots B(x_1), \exists xB(x)$$

All leaves of the above tree $\mathbf{T}_A$ are axioms, what means that we proved

$$\vdash_{QRS} (\exists x(A(x) \cap B(x)) \Rightarrow (\exists x A(x) \cap \exists x B(x))).$$

We construct now, as the last example, a decomposition tree $\mathbf{T}_A$ of the formula $((\exists x A(x) \cap \exists x B(x)) \Rightarrow \exists x(A(x) \cap B(x)))$.

**Example 10.6**

*The decomposition tree of the formula*

$$((\exists x A(x) \cap \exists x B(x)) \Rightarrow \exists x(A(x) \cap B(x)))$$

*is the following.*

$$\mathbf{T}_A$$

$$((\exists x A(x) \cap \exists x B(x)) \Rightarrow \exists x(A(x) \cap B(x)))$$

$$| \ (\Rightarrow)$$

$$\neg(\exists x A(x) \cap \exists x B(x)) \exists x(A(x) \cap B(x))$$

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

$$\neg\exists x A(x), \neg\exists x B(x), \exists x(A(x) \cap B(x))$$

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

$$\forall x \neg A(x), \neg\exists x B(x), \exists x(A(x) \cap B(x))$$

$$| \ (\forall)$$

$$\neg A(x_1), \neg\exists x B(x), \exists x(A(x) \cap B(x))$$

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

$$\neg A(x_1), \forall x \neg B(x), \exists x(A(x) \cap B(x))$$

$$| \ (\forall)$$

$$\neg A(x_1), \neg B(x_2), \exists x(A(x) \cap B(x))$$

*By the reasoning similar to the reasonings in the previous examples we get that* $x_1 \neq x_2$

$$| \ (\exists)$$

$$\neg A(x_1), \neg B(x_2), (A(t_1) \cap B(t_1)), \exists x(A(x) \cap B(x))$$

*where* $t_1$ *is the first term in the sequence (10.11), such that* $(A(t_1) \cap B(t_1))$ *does not appear on the tree above* $\neg A(x_1), \neg B(x_2), (A(t_1) \cap B(t_1)), \exists x(A(x) \cap B(x))$ *Observe, that it is possible that* $t_1 = x_1$, *as* $(A(x_1) \cap B(x_1))$ *does not appear on the tree above. By the definition of the*

*sequence (10.11), $x_1$ is placed somewhere in it, i.e. $x_1 = t_i$, for certain $i \geq 1$. For simplicity, we assume that $t_1 = x_1$ and get the sequence:*

$$\neg A(x_1), \neg B(x_2), (A(x_1) \cap B(x_1)), \exists x(A(x) \cap B(x))$$

$$\bigwedge(\cap)$$

$$\neg A(x_1), \neg B(x_2), \quad \neg A(x_1), \neg B(x_2),$$
$$A(x_1), \exists x(A(x) \cap B(x)) \quad B(x_1), \exists x(A(x) \cap B(x))$$
$$\textit{Axiom} \qquad | \ (\exists)$$
$$\neg A(x_1), \neg B(x_2), B(x_1),$$
$$(A(x_2) \cap B(x_2)), \exists x(A(x) \cap B(x))$$

*where $x_2 = t_2$ ($x_1 \neq x_2$) is the first term in the sequence (10.11), such that $(A(x_2) \cap B(x_2))$ does not appear on the tree above $\neg A(x_1), \neg B(x_2), (B(x_1), (A(x_2) \cap B(x_2)), \exists x(A(x) \cap B(x))$. We assume that $t_2 = x_2$ for the reason of simplicity.*

$$\bigwedge(\cap)$$

$$\neg A(x_1), \quad \neg A(x_1),$$
$$\neg B(x_2), \quad \neg B(x_2),$$
$$B(x_1), A(x_2), \quad B(x_1), B(x_2),$$
$$\exists x(A(x) \cap B(x)) \quad \exists x(A(x) \cap B(x))$$
$$| \ (\exists) \qquad \textit{Axiom}$$
$$\dots$$
$$\bigwedge(\cap)$$
$$\dots$$
$$| \ (\exists)$$
$$\dots$$
$$| \ (\exists)$$
$$\textit{infinite branch}$$

The above decomposition tree $\mathbf{T}_A$ contains an infinite branch what means that

$$\nvdash_{QRS} \ ((\exists x A(x) \cap \exists x B(x)) \Rightarrow \exists x(A(x) \cap B(x))).$$

## 10.3 Proof of QRS Completeness

Our main goal is to prove the Completeness Theorem for **QRS**. The proof of completeness theorem presented here is due to Rasiowa and Sikorski (1961), as is the proof system **QRS**. We adopted their proof to propositional case in Chap. 6. The completeness proofs, in the propositional case and in predicate case, are constructive as they are based on a direct construction of a counter model for any unprovable formula. The construction of the counter model for the unprovable formula A uses the decomposition tree $\mathbf{T}_A$. We call such constructed counter model a counter model determined by the tree $\mathbf{T}_A$. Rasiowa-Sikorski type of constructive proofs of counter models determined by the tree decomposition trees relay heavily of the notion of a *strong soundness*. We define it here (Definition 10.8), adopting Chap. 4 general definition to our case.

Given a first order language $\mathcal{L}$ (10.1) with the set $VAR$ of variables and the set $\mathcal{F}$ of formulas. We define, after Chap. 8 a notion of a model and a counter-model of a formula $A$ of $\mathcal{L}$ and then extend it to the set $\mathcal{F}^*$ establishing the **semantics** for **QRS**.

**Definition 10.2 (Model)**

*A structure $\mathcal{M} = [M, I]$ is called a model of $A \in \mathcal{F}$ if and only if*

$$(\mathcal{M}, v) \models A$$

*for all assignments $v : VAR \longrightarrow M$.*
*We denote it by*

$$\mathcal{M} \models A.$$

*$M$ is called the universe of the model, $I$ the interpretation.*

**Definition 10.3 (Counter – Model)**

*A structure $\mathcal{M} = [M, I]$ is called a counter-model of $A \in \mathcal{F}$ if and only if there is $v : VAR \longrightarrow M$, such that*

$$(\mathcal{M}, v) \not\models A.$$

*We denote it by*

$$\mathcal{M} \not\models A.$$

The definition of the first order logic tautology is the following.

**Definition 10.4 (Tautology)**

*For any $A \in \mathcal{F}$, $A$ is called a (predicate) tautology and denoted by*

$$\models A$$

*if and only if all structures* $\mathcal{M} = [M, I]$ *are models of A, i.e.*

$$\models A \quad \textit{if and only if} \quad \mathcal{M} \models A$$

*for all structures* $\mathcal{M} = [M, I]$ *for* $\mathcal{L}$.

Directly from the above definition we get the following, simple fact.

**Fact 10.1 (Counter Model)**

*For any* $A \in \mathcal{F}$, *A is not a tautology* ($\not\models A$) *if and only if there is a counter – model* $\mathcal{M} = [M, I]$ *of A, i.e. we can define* $M, I$, *and* $v$ *such that* $([M, I], v) \not\models A$.

**Definition 10.5**

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

$$\delta_\Gamma$$

*we understand any disjunction of all formulas of* $\Gamma$.

**Definition 10.6 (QRS Semantics)**

*A structure* $\mathcal{M} = [M, I]$ *for* $\mathcal{L}$ *is called a* **model** *of a* $\Gamma \in \mathcal{F}^*$ *and denoted by*

$$\mathcal{M} \models \Gamma$$

*if and only if*

$$\mathcal{M} \models \delta_\Gamma.$$

*The sequence* $\Gamma$ *is a predicate tautology if and only if the formula* $\delta_\Gamma$ *is a predicate tautology, i.e.*

$$\models \Gamma \quad \textit{if and only if} \quad \models \delta_\Gamma.$$

Our goal now is to prove the completeness theorem for **QRS**. The correctness of the proof we present depends on the strong soundness of the rules of inference of rules of inference defined as follows.

**Definition 10.7 (Strongly Sound Rules)**

*Given a predicate language (10.1) proof system* $S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R})$ *An inference rule* $r \in \mathcal{R}$ *of the form*

$$(r) \qquad \frac{P_1 \;;\; P_2 \;;\; \ldots \;;\; P_m}{C}$$

*is* **strongly sound** *if the following condition holds for and structure* $\mathcal{M} = [M, I]$ *for* $\mathcal{L}$.

$$\mathcal{M} \models \{P_1, P_2, .P_m\} \quad \textit{if and only if} \quad \mathcal{M} \models C. \tag{10.12}$$

We say it less formally that a rule (r) is **strongly sound** if the conjunction of its premisses is logically equivalent with the conclusion, i.e.

$$P_1 \cap P_2 \cap \ \ldots \ \cap P_m \equiv C. \tag{10.13}$$

**Definition 10.8 (Strongly Sound System)**

*A predicate language (10.1) proof system* $S = (\mathcal{L}, \mathcal{E}, LA, \mathcal{R})$ *is* **strongly sound** *if and only if all logical axioms* $LA$ *are tautologies and all its rules of inference* $r \in \mathcal{R}$ *are* **strongly sound**.

**Theorem 10.2 (Strong Soundness)**

*The proof system* **QRS** *(10.8) is* **strongly sound**.

**Proof**
We have already proved in Chap. 6 strong soundness of the propositional rules. The quantifiers rule are strongly sound by straightforward verification and is left as an exercise.

The strong soundness property is stronger then soundness property, hence also the following holds.

**Theorem 10.3 (Soundness Theorem)**

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

$$\textit{if} \quad \vdash_{QRS} \Gamma, \quad \textit{then} \quad \models \Gamma.$$

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

$$\textit{if} \quad \vdash_{QRS} A, \quad \textit{then} \quad \models A.$$

**Theorem 10.4 (Completeness Theorem)**

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

$$\vdash_{QRS} \Gamma \quad \textit{if and only if} \quad \models \Gamma.$$

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

$$\vdash_{QRS} A \quad \textit{if and only if} \quad \models A.$$

**Proof**
We have to prove the inverse implication to the soundness Theorem 10.3. We need to prove the formula $A$ case only because the case of a sequence $\Gamma$ can be reduced to the formula case. Namely, the disjunction of all formulas in $\Gamma$. I.e. we prove the implication:

$$\text{if} \quad \models A, \quad \text{then} \quad \vdash_{QRS} A.$$

We do it, as in the propositional case, by proving the opposite implication

$$\text{if} \quad \nvdash_{QRS} \ A \quad \text{then} \quad \not\models A.$$

This means that we want prove that for any formula $A$, unprovability of $A$ in **QRS** ($\nvdash_{QRS} \ A$ ), allows us to define its counter-model. The counter-model is determined, as in the propositional case, by the decomposition tree $\mathbf{T}_A$. By Theorem 10.1 each formula $A$, generates its unique decomposition tree $\mathcal{T}_A$ and $A$ has a proof only if this tree is finite and all its end sequences (leaves) are axioms. Moreover, it says that we have two cases to consider:

**(C1)** the tree $\mathbf{T}_A$ is **finite** and contains a leaf which is not axiom, or

**(C2)** the tree $\mathbf{T}_A$ is **infinite**.

We will show how to construct a counter-model for $A$ in both cases: a counter-model determined by a non-axiom leaf of the decomposition tree $\mathbf{T}_A$, or a counter-model determined by an infinite branch of $\mathbf{T}_A$.

**Proof** in case **(C1)**: $\mathbf{T}_A$ is **finite** and contains a non- axiom leaf.

Before describing a general method of constructing the counter-model determined by the decomposition tree $\mathcal{T}_A$ we describe it, as an example, for a case of a formula

$$(\exists x A(x) \Rightarrow \forall x A(x)),$$

and its **particular case**

$$(\exists x(P(x) \cap R(x,y)) \Rightarrow \forall x(P(x) \cap R(x,y))) \tag{10.14}$$

for $P$, $R$ one and two argument predicate symbols, respectively.

We construct the counter model for the formula (10.14) as follows.

First we build its decomposition tree:

$$\mathbf{T}_A$$

$$(\exists x(P(x) \cap R(x,y)) \Rightarrow \forall x(P(x) \cap R(x,y)))$$

$$| \ (\Rightarrow)$$

$$\neg\exists x(P(x) \cap R(x,y)), \forall x(P(x) \cap R(x,y))$$

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

$$\forall x\neg(P(x) \cap R(x,y)), \forall x(P(x) \cap R(x,y))$$

$$| \ (\forall)$$

$$\neg(P(x_1) \cap R(x_1,y)), \forall x(P(x) \cap R(x,y))$$

where $x_1$ is a first free variable in (10.11) such that $x_1$ does not appear in $\forall x\neg(P(x)\cap R(x,y)), \forall x(P(x)\cap R(x,y))$

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

$$\neg P(x_1), \neg R(x_1,y), \forall x(P(x)\cap R(x,y))$$

$$\mid (\forall)$$

$$\neg P(x_1), \neg R(x_1,y), (P(x_2)\cap R(x_2,y))$$

where $x_2$ is a first free variable in the sequence (10.11) such that $x_2$ does not appear in $\neg P(x_1), \neg R(x_1,y), \forall x(P(x)\cap R(x,y))$, the sequence (10.11) is one-to- one, hence $x_1 \neq x_2$

$$\bigwedge(\cap)$$

$$\neg P(x_1), \neg R(x_1,y), P(x_2) \qquad \neg P(x_1), \neg R(x_1,y), R(x_2,y)$$

$x_1 \neq x_2$, Non-axiom $\qquad$ $x_1 \neq x_2$, Non-axiom

There are two non-axiom leaves. In order to define a counter-model for (10.14) determined by the tree $\mathbf{T}_A$ we need to chose only one of them. Let's choose the leaf

$$L_A = \neg P(x_1), \neg R(x_1,y), P(x_2). \tag{10.15}$$

We use the **non-axiom leaf** $L_A$ to define a structure $\mathcal{M} = [M, I]$ and an assignment $v$, such that

$$(\mathcal{M}, v) \not\models A.$$

Such defined $\mathcal{M}$ is called a **counter − model** determined by the tree $\mathbf{T}_A$.

We take a the universe of $\mathcal{M}$ the set $\mathbf{T}$ of all terms of our language $\mathcal{L}$, i.e. we put

$$M = \mathbf{T}.$$

We define the interpretation $I$ as follows. For any predicate symbol $Q \in \mathbf{P}, \#Q = n$ we put that $Q_I(t_1, \ldots t_n)$ is **true** (holds) for terms $t_1, \ldots t_n$ if and only if the negation $\neg Q_I(t_1, \ldots t_n)$ of the formula $Q(t_1, \ldots t_n)$ appears on the leaf $L_A$ and $Q_I(t_1, \ldots t_n)$ is **false** (does not hold) for terms $t_1, \ldots t_n$ otherwise.

For any functional symbol $f \in \mathbf{F}, \#f = n$ we put $f_I(t_1, \ldots t_n) = f(t_1, \ldots t_n)$.

It is easy to see that in particular case of our non-axiom leaf (10.15)

$$L_A = \neg P(x_1),\ \neg R(x_1,y),\ P(x_2)$$

$P_I(x_1)$ is true for $x_1$, and not true for $x_2$. $R_I(x_1,y)$ is true (holds) holds for $x_1$ any for any $y \in VAR$.

We define the assignment $v : VAR \longrightarrow T$ as *identity*, i.e., we put $v(x) = x$ for any $x \in VAR$.

Obviously, for such defined structure $[M, I]$ and the assignment $v$ we have that

$$([\mathbf{T}, I],\ v) \models P(x_1), \quad ([\mathbf{T}, I],\ v) \models R(x_1, y), \quad \text{and} \quad ([\mathbf{T}, I],\ v) \not\models P(x_2).$$

We hence obtain that

$$([\mathbf{T}, I],\ v) \not\models \neg P(x_1), \neg R(x_1, y), P(x_2).$$

This proves that such defined structure $[\mathbf{T}, I]$ is a counter model for a non-axiom leaf (10.15). By the strong soundness of QRS (Theorem 10.2) the structure $\mathcal{M} = [\mathbf{T}, I]$ is also a counter-model for the formula (10.14), i.e. we proved that

$$\not\models (\exists x(P(x) \cap R(x, y)) \Rightarrow \forall x(P(x) \cap R(x, y))).$$

**C1: General Method**
Let $A$ be any formula such that $\nvdash_{QRS} A$.

Let $\mathcal{T}_A$ be a decomposition tree of $A$. By the fact that $\nvdash_{QRS}$ and **C1**, the tree $\mathcal{T}_A$ is finite and has a *non axiom leaf*

$$L_A \subseteq LT^*. \tag{10.16}$$

By definition, the leaf $L_A$ contains only atomic formulas and negations of atomic formulas.

We use the **non-axiom leaf** $L_A$ (10.16) to define a structure $\mathcal{M} = [M, I]$ an assignment $v : VAR \longrightarrow M$, such that $(\mathcal{M}, v) \not\models A$. Such defined structure $\mathcal{M}$ is called a **counter − model** determined by the tree $\mathbf{T}_A$.

**Structure $\mathcal{M}$ Definition** (10.17)

Given $L_A$. We define a structure

$$\mathcal{M} = [M, I] \tag{10.18}$$

and an assignment $v : VAR \longrightarrow M$ as follows.

**1.** We take a the universe of $\mathcal{M}$ the set $\mathbf{T}$ of all terms of our language $\mathcal{L}$' i.e. we put

$$M = \mathbf{T}.$$

**2.** For any predicate symbol $Q \in \mathbf{P}, \#Q = n$,

$$Q_I \subseteq \mathbf{T}^n$$

is such that $Q_I(t_1, \ldots t_n)$ holds (is true) for terms $t_1, \ldots t_n$ if and only if the negation $\neg Q(t_1, \ldots t_n)$ of the formula $Q(t_1, \ldots t_n)$ appears on the leaf $L_A$ and $Q_I(t_1, \ldots t_n)$ does not hold (is false) for terms $t_1, \ldots t_n$ otherwise.

**3.** For any constant $c \in \mathbf{C}$, we put $c_I = c$, for any variable $x$, $x_I = x$.
For any functional symbol $f \in \mathbf{F}, \#f = n$,

$$f_I : \mathbf{T}^n \longrightarrow \mathbf{T}$$

is identity function, i.e. we put

$$f_I(t_1, \ldots t_n) = f(t_1, \ldots t_n)$$

for all $t_1, \ldots t_n \in \mathbf{T}$.

**4.** We define the assignment $v : VAR \longrightarrow \mathbf{T}$ as *identity*, i.e. we put for all $x \in VAR$ $v(x) = x$.

Obviously, for such defined structure $[\mathbf{T}, I]$ and the assignment $v$ we have that

$$([\mathbf{T}, I],\ v) \not\models\ P \quad \text{if formula } P \text{ appears in } \quad L_A,$$

$$([\mathbf{T}, I],\ v) \models\ P \quad \text{if formula } \neg P \text{ appears in } \quad L_A.$$

This proves that the structure $\mathcal{M} = [\mathbf{T}, I]$ and assignment $v$ defined by (10.18) are such that

$$([\mathbf{T}, I], v) \not\models\ L_A.$$

By the **strong soundness** (Theorem 10.2) of **QRS**

$$([\mathbf{T}, I], v) \not\models\ A.$$

This proves $\mathcal{M} \not\models\ A$ and we proved that

$$\not\models\ A.$$

This ends the proof of the case **C1**.

**Proof** in case **(C2)**: $\mathbf{T}_A$ is **infinite**.

The case of the **infinite tree** is similar, even if a little bit more complicated. Observe first that the rule $(\exists)$ is the t rule of inference (decomposition) which can "produces" an infinite branch. We first show how to construct the counter-model in the case of the simplest application of this rule, i.e. in the case of the formula

$$\exists x P(x)$$

where $P$ is an one argument relational symbol. All other cases are similar to this one. The infinite branch $\mathcal{B}_A$ in this case consists of all elements of the decomposition tree:

$$\mathbf{T}_A$$

$$\exists x P(x)$$

$$| \ (\exists)$$

$$P(t_1), \exists x P(x)$$

where $t_1$ is the first term in the sequence (10.11), such that $P(t_1)$ does not appear on the tree above $P(t_1), \exists x P(x)$

$$| \ (\exists)$$

$$P(t_1), P(t_2), \exists x P(x)$$

where $t_2$ is the first term in the sequence (10.11), such that $P(t_2)$ does not appear on the tree above $P(t_1), P(t_2), \exists x P(x)$, i.e. $t_2 \neq t_1$

$$| \ (\exists)$$

$$P(t_1), P(t_2), P(t_3), \exists x P(x)$$

where $t_3$ is the first term in the sequence (10.11), such that $P(t_3)$ does not appear on the tree above $P(t_1), P(_2), P(t_3), \exists x P(x)$, i.e. $t_3 \neq t_2 \neq t_1$

$$| \ (\exists)$$

$$P(t_1), P(t_2), P(t_3), P(t_4), \exists x P(x)$$

$$| \ (\exists)$$

$$\ldots..$$

$$| \ (\exists)$$

$$\ldots..$$

The infinite branch of $\mathbf{T}_A$, written from the top, in order of appearance of formulas is

$$\mathcal{B}_A = \{\exists x P(x), \ P(t_1), \ A(t_2), \ P(t_2), \ P(t_4), \ldots..\}$$

where $t_1, t_2, \ldots.$ is a one - to one sequence (10.11) of all elements of the set $\mathbf{T}$ of all terms.

This means that the infinite branch $\mathcal{B}$ contains with the formula $\exists x P(x)$ all its instances $P(t)$, for all terms $t \in \mathbf{T}$.

We define the structure $\mathcal{M} = [M, I]$ and valuation $v$ following the Definition 10.17. We take as the universe $M$ the set $\mathbf{T}$ of all terms, and now in our case we define $P_I$ as follows: $P_I(t)$ holds if $\neg P(t) \in \mathcal{B}_A$ and $P_I(t)$ does not hold if $P(t) \in \mathcal{B}_A$.

It is easy to see that for any formula $P(t) \in \mathcal{B}$,

$$([T, I], v) \not\models P(t).$$

But the $A(t) \in \mathcal{B}$ are all instances $\exists x A(x)$, hence

$$([T, I], v) \not\models \exists x A(x).$$

**C1: General Method**
Let $A$ be any formula such that $\not\vdash_{QRS} A$.

Let $\mathcal{T}_A$ be an infinite decomposition tree of a formula $A$. Let $\mathcal{B}_A$ the infinite branch of $\mathbf{T}_A$, written from the top, in order of appearance of sequences $\Gamma \in \mathcal{F}^*$ on it, where $\Gamma_0 = A$.

$$\mathcal{B}_A = \{\Gamma_0, \ \Gamma_1, \ \Gamma_2, \ \ldots \ \Gamma_i, \ \Gamma_{i+1}, \ \ldots\} \tag{10.19}$$

We define a set $L\mathcal{F} \subseteq \mathcal{F}$ of all *indecomposable* formulas appearing in at least one $\Gamma_i$, $i \leq j$, i.e.

$$L\mathcal{F} = \{B \in LT : \ \text{there is } \Gamma_i \in \mathcal{B}_A, \text{ such that } B \text{ is in } \Gamma_i\}. \tag{10.20}$$

Note, that the following holds.
(1) if $i \leq i'$ and an indecomposable formula appears in $\Gamma_i$, then it also appears in $\Gamma_{i'}$.
(2) Since none of $\Gamma_i$, is an axiom (10.6), for every atomic formula (10.2) $P \in A\mathcal{F}$, at most one of the formulas $P$ and $\neg P$ is in $L\mathcal{F}$ (10.20).

**Counter Model Definition**
Let $\mathbf{T}$ be the set of all terms. We define the structure $\mathcal{M} = [\mathbf{T}, I]$ and valuation $v$ in the set $\mathbf{T}$ as in the Definition 10.17, with the interpretation of predicates $Q \in \mathbf{Q}$ defined as follows.

For any predicate symbol $Q \in \mathbf{P}, \#Q = n$, $Q_I \subseteq \mathbf{T}^n$ is such that:

(1) $Q_I(t_1, \ldots t_n)$ does not hold (is false) for terms $t_1, \ldots t_n$ if and only if

$$Q_I(t_1, \ldots t_n) \in L\mathcal{F}$$

and
(2) $Q_I(t_1, \ldots t_n)$ does holds (is true) for terms $t_1, \ldots t_n$ if and only if

$$Q_I(t_1, \ldots t_n) \notin L\mathcal{F}.$$

This proves that the structure $\mathcal{M} = [\mathbf{T}, I]$ is such that

$$\mathcal{M} \not\models L\mathcal{F}. \tag{10.21}$$

To prove that $\not\models A$ it suffices that

$$\mathcal{M} \not\models A. \tag{10.22}$$

For this purpose we first introduce, for any formula $A \in \mathcal{F}$, an inductive definition of the order $ord\ A$ of the formula A.
(1) If $A \in A\mathcal{F}$, then $ord\ A = 1$.
(2) If $ord\ A = n$, then $ord\ \neg A = n + 1$. (3) If $ord\ A \leq n$ and $ord\ B \leq n$, then $ord\ (A \cup B) = ord\ (A \cap B) = ord\ (A \Rightarrow B) = n + 1$.
(4) If $ord\ A(x) = n$, then $ord\ \exists x A(x) = ord\ \forall x A(x) = n + 1$.

We conduct the proof of (10.23) by contradiction. Suppose that (10.23) does not hold, i.e. assume that

$$\mathcal{M} \models A. \tag{10.23}$$

Consider now a set $M\mathcal{F}$ of all formulas $B$ appearing in one of the sequences $\Gamma_i$ of the branch $\mathcal{B}_A$, such that

$$\mathcal{M} \models B. \tag{10.24}$$

We write the set $M\mathcal{F}$ formally as follows.

$$M\mathcal{F} = \{B \in \mathcal{F} : \text{ for some } \Gamma_i \in \mathcal{B}_A,\ B \text{ is in } \Gamma_i \text{ and } \mathcal{M} \models B\}. \tag{10.25}$$

Observe that by assumption (10.23) and the Definition (10.25), the formula $A$ is in $M\mathcal{F}$ and hence $M\mathcal{F} \neq \emptyset$.

Let $B'$ be a formula in $M\mathcal{F}$ such that $ord\ B' \leq ord\ B$ for every $B \in M\mathcal{F}$. There exists $\Gamma_i \in\in \mathcal{B}_A$ that is of the form $\Gamma', B', \Delta$ with an indecomposable $\Gamma'$.
We have that $B'$ can not be of the form

$$\neg\exists x A(x) \quad \text{or} \quad \neg\forall x A(x) \tag{10.26}$$

for if (refn-Q) is in $M\mathcal{F}$, then also formula $\forall x \neg A(x)$ or $\exists x \neg A(x)$ is in $M\mathcal{F}$ and the orders of the two formulas are equal.

We carry the same order argument and show that $B'$ can not be of the form

$$(A \cup B), \neg(A \cup B), (A \cap B), \neg(A \cap B), (A \Rightarrow B), \neg(A \Rightarrow B), \neg\neg A, \forall x A(x). \tag{10.27}$$

The formula $B'$ can't be of the form

$$\exists x B(x) \tag{10.28}$$

since then there exists term $t$ and $j$ such that $i \leq j$, $B'(t)$ appears in $\Gamma_j$ and the formula $B(t)$ satisfies (10.24). Thus $B(t) \in M\mathcal{F}$ and $ordB(t) < ordB'$. This contradicts the definition of $B'$.

Since $B'$ is not of the form (10.26), (10.27), (10.28), $B'$ is indecomposable. Thus $B' \in L\mathcal{F}$ (10.20), and consequently by (10.21),

$$\mathcal{M} \not\models B'.$$

On the other hand $B'$ by definition is in the set $M\mathcal{F}$ and hence is one o the formulas satisfying (10.24), i.e.

$$\mathcal{M} \not\models B'.$$