CHAPTER 13

Gentzen Style Proof System for Classical Predicate Logic - The System QRS

Part One

1 System QRS Definition

Let $\mathcal{F}$ denote a set of formulas of a Predicate (first Order) Logic Language

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

for **P, F, C** countably infinite sets of predicate, functional, and constant symbols respectively.

The rules of inference of our system **QRS** will operate, as in the propositional case, on *finite sequences of formulas*, i.e. elements of $\mathcal{F}^*$, instead of just plain formulas $\mathcal{F}$, as in Hilbert style formalizations. We will denote the sequences of formulas by Γ, Δ, Σ, with indices if necessary.

that the truth assignment v makes it true if and only if it makes the formula of the form of thetrue.

The intuitive meaning of a sequence $\Gamma \in \mathcal{F}^*$ is that it represents a disjunction of all formulas of Γ, i.e. if Γ is a sequence

$$A_1, A_2, ..., A_n$$

then by δ_Γ we will understand the disjunction of all formulas of Γ.

As we know, the disjunction in classical logic is commutative, i.e., for any formulas A, B, C, $A \cup (B \cup C) \equiv (A \cup B) \cup C$, we w will denote any of those formulas by $A \cup B \cup C = \delta_{\{A,B,C\}}$. Similarly, we will write $\delta_\Gamma = A_1 \cup A_2 \cup ..., \cup A_n$.

The sequence Γ is said to be *satisfiable* (*falsifiable*) if the formula $\delta_\Gamma = A_1 \cup A_2 \cup ..., \cup A_n$ is satisfiable (falsifiable).

The sequence Γ is said to be *a tautology* if the formula $\delta_\Gamma = A_1 \cup A_2 \cup ..., \cup A_n$ is a tautology.

The system **QRS** consists of *one axiom* and *eleven rules* of inference. They form two groups. First is similar to the propositional case and called propositional connectives group. Each rule of this group introduces a new logical connective or its negation, so we will name them, as in the propositional case: $(\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg \Rightarrow)$, and $(\neg\neg)$. The second group deals with the quantifiers. It consists of four rules. Two of them introduce the universal and existential quantifiers, and are named $(\forall)$ and $(\exists)$, respectively. The two others correspond to the De Morgan Laws and deal with the negation of the universal and existential quantifiers, and ere named $(\neg\forall)$ and $(\neg\exists)$, respectively.

As the *axiom* we adopt, as in propositional case, any sequence which contains any formula and its negation, i.e any sequence of the form

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

or of the form

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

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

We will denote the axioms by

$$\mathcal{AX}^*.$$

The proof system

$$\mathbf{QRS} = (\mathcal{F}^*, \mathcal{AX}^*, (\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg \Rightarrow), (\neg\neg), (\neg\forall), (\neg\exists), (\forall), (\exists))$$

will be called a *Gentzen- style formalization* of classical predicate calculus.

In order to define the rules of inference of **QRS** we need to introduce some definitions. They are straightforward modification of the corresponding definitions for the propositional logic.

We will form now, as in the propositional case, a special subset $\mathcal{LIT} \subseteq \mathcal{F}$ of formulas, called a set of all *literals*, which is defined now as follows.

$$\mathcal{LIT} = \{A \in \mathcal{F} : A \in \mathcal{AF}\} \cup \{\neg A \in \mathcal{F} : A \in \mathcal{AF}\},$$

where $\mathcal{AF} \subseteq \mathcal{F}$ is the set of all atomic (elementary) formulas of the first order language, i.e. $\mathcal{AF} = \{P(t_1, ..., t_n) : P \in \mathbf{P}$ is any n-argument predicate symbol, and $t_i \in T$ are terms $\}$.

The elements of the first set of the above union are called *positive literals* and the elements of the second set of the above union are called *negative literals*. I.e atomic (elementary) formulas are called positive literals and the negation of an atomic (elementary) formula is called a negative literal.

Indecomposable formulas

Literals are also called the indecomposable formulas.

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 exactly the same notation as in the propositional case. We will denote by $\Gamma^{'}$, $\Delta^{'}$, $\Sigma^{'}$ finite sequences (empty included) formed out of *literals* i.e. out of the elements of $\mathcal{LIT}$ i.e. we assume that $\Gamma^{'}$, $\Delta^{'}$, $\Sigma^{'} \in \mathcal{LIT}^*$.

We will denote by Γ, Δ, Σ the elements of $\mathcal{F}^*$ i.e the finite sequences (empty included) formed out of elements of $\mathcal{F}$.

We define the inference rules of **QRS** as follows.

Group 1: Propositional Inference 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 $\Gamma^{'} \in \mathcal{F}^*, \Delta \in \mathcal{F'}^*, A, B \in \mathcal{F}$.

Group 2: Quantifiers Rules

(∃)

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

where t is an arbitrary term.

(∀)

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

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

(¬∀)

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

(¬∃)

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

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

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 (∀) 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 (∀) are fully indicated.

We define the notion of a *formal proof* in **QRS** as in any proof system, i.e., by a formal proof of a sequence Γ in the proof system

$$\mathbf{QRS} = (\mathcal{F}^*, \mathcal{AX}^*, (\cup), (\neg\cup), (\cap), (\neg\cap), (\Rightarrow), (\neg \Rightarrow), (\neg\neg), (\neg\exists), (\neg\forall), \exists), (\forall))$$

we understand any sequence $\Gamma_1\Gamma_2....\Gamma_n$ of sequences of formulas (elements of $\mathcal{F}^*$, such that $\Gamma_1 \in \mathcal{AX}^*$, $\Gamma_n = \Gamma$, and for all i $(1 < i \leq n)$ $\Gamma_i \in \mathcal{AX}^*$, or Γ_i is a conclusion of one of the inference rules of **QRS** with all its premisses placed in the sequence $\Gamma_1\Gamma_2....\Gamma_{i-1}$.

As the proof system under consideration is fixed, we will write, as usual,

$$\vdash \Gamma$$

to denote that Γ has a formal proof in **QRS**.

As the proofs in **QRS** are sequences (definition of the formal proof) of sequences of formulas (definition of **GQ**) we will not use "," to separate the steps of the proof, i.e. will write the sequence the formal proof as a sequence $\Gamma_1\Gamma_2....\Gamma_n$ instead of $\Gamma_1, \Gamma_2, ..., \Gamma_n$, but usually we will use, as in the propositional case, the *proof trees* to represent the formal proofs. The *leafs* of the proof-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 sequence (formula). We will 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 the proof of a *theorem* from 3 *axioms* used subsequently the rules $(\cap), (\exists), (\forall), (\cap), (\neg\exists), (\neg\neg)$, and $(\Rightarrow)$, we will represent it as the following tree

$$Sequence(Formula)$$

$$\mid (\Rightarrow)$$

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

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

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

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

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

$$\bigwedge(\cap)$$

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

$$\mid (\forall) \qquad \mid (\exists)$$

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

$$\bigwedge(\cap)$$

$$axiom \qquad axiom$$

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. It is easy to define the tree-proofs precisely, as well as a general transformation procedure between the tree and the sequence form of the proofs, but we will explain it here on few examples only.

2 QRS Decomposition Trees

Given a formula $A \in \mathcal{F}$, we define its decomposition tree $\mathcal{T}_A$ in a similar way as in the propositional case. Observe that the inference rules of **QRS** can be divided in two groups: propositional 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. 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

$$t_1, t_2, ..., t_n, ... \tag{1}$$

Observe, that by the definition, all free variables are terms, hence all free variables appear in the sequence 1. Let Γ be a sequence on the tree with $\forall$ as a main connective, i.e. Γ 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 2.1 ($\forall$) *: x is the first free variable in the sequence 1 such that x does not appear in any formula in $\Gamma^{'}, \forall x A(x), \Delta$.*

Observe, that the condition 2.1 corresponds to the restriction put on the application of the rule $(\forall)$.

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

Condition 2.2 ($\exists$) *:*

t is the first term in the sequence 1 such that the formula $A(t)$ does not appear in any sequence which is placed above $\Gamma^{'}, A(t), \Delta$ on the tree.

The fact that the sequence 1 is one- to - one and the fact that, by the conditions 2.1 and 2.2, 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 2.1 (Uniqueness) *For any formula $A \in \mathcal{F}$, its decomposition tree $\mathcal{T}_A$ is unique. If $\mathcal{T}_A$ is finite and all its leaves are axioms, then $\vdash A$ and $\mathcal{T}_A$ is a tree-proof of A in* **QRS**. *If $\mathcal{T}_A$ is finite and contains a non-axiom leaf or is infinite, then $\not\vdash A$.*

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.

The decomposition tree of the de Morgan Law $(\neg\forall x A(x) \Rightarrow \exists x \neg A(x))$ is the following.

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

$$\mid (\Rightarrow)$$

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

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

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

$$\mid (\forall)$$

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

where x_1 is a first free variable in the sequence 1 such that x_1 does not appear in $\forall x A(x), \exists x \neg A(x)$

$$\mid (\exists)$$

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

where x_1 is the first term (variables are terms) in the sequence 1 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 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.

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

The decomposition tree of $(\forall x A(x) \Rightarrow \exists x A(x))$ is the following.

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

$$\mid (\Rightarrow)$$

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

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

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

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

$$\mid (\exists)$$

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

where t_1 is the first term in the sequence 1, such that $\neg A(t_1)$ does not appear on the tree above $\neg A(t_1), \exists x A(x), \exists x \neg A(x)$

$$\mid (\exists)$$

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

where t_1 is the first term in the sequence 1, 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

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

The decomposition tree of $(\exists x A(x) \Rightarrow \forall x A(x))$ is the following.

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

$$\mid (\Rightarrow)$$

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

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

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

$$\mid (\forall)$$

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

where x_1 is a first free variable in 1 such that x_1 does not appear in $\forall x \neg A(x), \forall x A(x)$

$$\mid (\forall)$$

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

where x_2 is a first free variable in 1 such that x_2 does not appear in $\neg A(x_1), \forall x A(x)$, the sequence 1 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

$$\not\vdash (\exists x A(x) \Rightarrow \forall x A(x)).$$

The decomposition tree of $\exists x A(x)$ is the following.

$$\exists x A(x)$$

$$\mid (\exists)$$

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

where t_1 is the first term in the sequence 1, such that $A(t_1)$ does not appear on the tree above $A(t_1), \exists x A(x)$

$$\mid (\exists)$$

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

where t_2 is the first term in the sequence 1, such that $A(t_2)$ does not appear on the tree above $A(t_1), A(t_2), \exists x A(x)$, i.e. $t_2 \neq t_1$

$$\mid (\exists)$$

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

where t_3 is the first term in the sequence 1, 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. $t_3 \neq t_2 \neq t_1$

$$\mid (\exists)$$

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

$$\mid (\exists)$$

.....

$$\mid (\exists)$$

.....

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

$$\not\vdash \exists x A(x).$$

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 $((\exists x A(x) \cap \exists x B(x)) \Rightarrow \exists x(A(x) \cap B(x)))$. The decomposition tree of the first formula is the following.

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

$$\mid (\Rightarrow)$$

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

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

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

$$\mid (\forall)$$

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

where x_1 is a first free variable in the sequence 1 such that x_1 does not appear in $\forall x\neg(A(x)\cap B(x)), (\exists xA(x)\cap\exists xB(x))$

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

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

$$\bigwedge(\cap)$$

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

$$\mid (\exists)$$

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

where t_1 is the first term in the sequence 1, 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 1, 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:

$$\mid (\exists)$$

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

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

$$\mid (\exists)$$

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

$$\mid (\exists)$$

$$...$$

$$\mid (\exists)$$

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

All leaves of the above tree are axioms, what means that

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

Let's now construct, as the last example, a decomposition tree of

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

We will adopt, as on the right branch of the above tree, the shorthand notation used on this branch instead of the reasoning performed on the left branch, when the reasoning is similar to the one presented above. The decomposition tree is the following.

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

$$| (\Rightarrow)$$

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

$$| (\neg\cap)$$

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

$$| (\neg\exists)$$

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

$$| (\forall)$$

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

$$| (\neg\exists)$$

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

$$| (\forall)$$

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

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 1, 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 1, 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:

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

$$\bigwedge (\cap)$$

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

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

where $x_2 = t_2$ $(x_1 \neq x_2)$ is the first term in the sequence 1, 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.

$$\bigwedge(\cap)$$

$$\neg A(x_1), \neg B(x_2), B(x_1), A(x_2), \exists x(A(x) \cap B(x))$$
$$\mid (\exists)$$
$$...$$
$$\bigwedge(\cap)$$
$$...$$
$$\mid (\exists)$$
$$...$$
$$\mid (\exists)$$
$$Infinite\ branch$$

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

The above decomposition tree contains an infinite branch what means that

$$\not\vdash ((\exists x A(x) \cap \exists x B(x)) \Rightarrow \exists x(A(x) \cap B(x))).$$