# On uniform proof-theoretic operational semantics for logic programming*

Ekaterina Komendantskaya† Vladimir Komendantsky‡

October 31, 2006

**Abstract**

We study a general proof-theoretic framework for logic programming, the so-called uniform proofs. We consider several logic programming systems derived in a natural way from logical proof systems (namely, from classical, intuitionistic, minimal, positive, relevance and paranormal sequent calculi). Our result is the construction of proof-theoretic logic programming systems for philosophically significant logics.
**Keywords:** logic programming, uniform proofs, classical logic, lattice relevance logic, paranormal logic, non-deterministic closure operator, non-deterministic substitution.
**MSC:** 03B35, 68T15, 03B10, 03B47, 03B53

## 1 Introduction

The concept of uniform proofs is one of the computational paradigms behind logic programming. It was introduced by Miller in mid-80's. There is an easily accessible paper [5] covering this subject. This concept is probably more familiar to logicians studying logic programming than to computer scientists studying the same field because, in the case of first-order logics, uniform proofs mirror the structural features of sequent calculi into the language of first-order Horn clauses. So, using uniform proofs one can do logic programming staying very close to sequent calculi (as opposed to the alternative resolution-based paradigm [4]).

There were some variations on Miller's account, including a modal extension of uniform proofs [1] and a linear logic extension [3]. Those extensions

*The paper is based on a talk given by the authors during UNILOG-05, 26 March–3 April, 2005, Montreux, Switzerland.

†Maths Department, University College Cork, Ireland; e.komendantskaya@mars.ucc.ie.

‡Boole Centre for Research in Informatics, University College Cork, Ireland; v.komendantsky@bcri.ucc.ie.

involved some changes made to the language of clauses (adding modalities, connectives, etc.). In the present paper we keep the language the same as it was in [5], but we modify the set of inference rules of the original system of logic programming in order to adapt it to other sequent calculi, in particular, to calculi of classical logic (Subsection 3.1), relevance logic (Subsection 3.2), paranormal logic and intuitionistic paranormal logic (Subsection 3.3). As a result, we get logic programming systems, where, for instance, one can derive his favourite law of excluded middle, or can reason in resource-conscious conditions, or can manage databases with inconsistent contents.

## 2 Operational semantics

In this section we present the operational semantics of uniform proofs in an almost identical fashion to [5]. Later we will reuse certain definitions from this section to produce other variations of proof-theoretic semantics.

Let $A$ denote a first-order atomic formula. Let a clause $D$ and a goal formula $G$ be given by the following mutually recursive definition:

$$\begin{array}{lcl} D & ::= & \bot \mid A \mid G \supset A \mid G \supset \bot \mid \forall x D \mid D_1 \wedge D_2 \ , \\ G & ::= & \bot \mid A \mid G_1 \wedge G_2 \mid G_1 \vee G_2 \mid \exists x G \mid D \supset G \ . \end{array}$$

Define a logic program to be a finite set (a conjunction) of clauses.

Let $\mathcal{P}$ be a logic program. Define a *closure of* $\mathcal{P}$, denoted $[\mathcal{P}]$, be the smallest set of formulas satisfying the following recursive conditions:

- $\mathcal{P} \subseteq [\mathcal{P}]$;
- if $D_1 \wedge D_2 \in [\mathcal{P}]$ then $D_1 \in [\mathcal{P}]$ and $D_2 \in [\mathcal{P}]$;
- if $\forall x D \in [\mathcal{P}]$ then $D\{x/t\} \in [\mathcal{P}]$ for all terms $t$.

Here $D\{x/t\}$ denotes the result of substituting the term $t$ for all free occurrences of the variable $x$ in $D$.

We use the expression $\mathcal{P} \vdash G$ to say that $G$ can be derived from $\mathcal{P}$, or that $G$ is an output of $\mathcal{P}$. Logical connectives will be interpreted as search instructions. Additionally, we will use the subscript notation for the derivability relation $\vdash$ to distinguish between different systems of these instructions. One can accept operational inference system $\widehat{\mathbf{I}}$ (originally denoted $\mathbf{O}'$ in Miller's notation), consisting of the so-called search instructions listed below:

$\widehat{\mathbf{I}}.1$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{I}}} A$ if $A \in [\mathcal{P}]$;

$\widehat{\mathbf{I}}.2$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{I}}} A$ if there is a formula $(G \supset A) \in [\mathcal{P}]$ and $\mathcal{P} \vdash_{\widehat{\mathbf{I}}} G$;

$\widehat{\mathbf{I}}.3$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{I}}} G_1 \vee G_2$ if $\mathcal{P} \vdash_{\widehat{\mathbf{I}}} G_1$ or $\mathcal{P} \vdash_{\widehat{\mathbf{I}}} G_2$;

$\widehat{\mathbf{I}}.4$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{I}}} G_1 \wedge G_2$ if $\mathcal{P} \vdash_{\widehat{\mathbf{I}}} G_1$ and $\mathcal{P} \vdash_{\widehat{\mathbf{I}}} G_2$;

$\widehat{\mathbf{I}}.5$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{I}}} \exists x G$ if there is some term $t$ such that $\mathcal{P} \vdash_{\widehat{\mathbf{I}}} G\{x/t\}$;

$\widehat{\mathbf{I}}.6$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{I}}} D \supset G$ if $\mathcal{P} \cup \{D\} \vdash_{\widehat{\mathbf{I}}} G$;

$\widehat{\mathbf{I}}.7$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{I}}} \perp$ if $\perp \in [\mathcal{P}]$;

$\widehat{\mathbf{I}}.8$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{I}}} \perp$ if there is a formula $(G \supset \perp) \in [\mathcal{P}]$ and $\mathcal{P} \vdash_{\widehat{\mathbf{I}}} G$;

$\widehat{\mathbf{I}}.9$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{I}}} G$ if $\mathcal{P} \vdash_{\widehat{\mathbf{I}}} \perp$;

We say that a goal $G$ has an $\widehat{\mathbf{I}}$-derivation from a program $\mathcal{P}$ if it has been derived from $\mathcal{P}$ using instructions $\widehat{\mathbf{I}}.1$–$\widehat{\mathbf{I}}.9$. We say that $G$ has an $\widehat{\mathbf{M}}$-derivation (respectively, $\widehat{\mathbf{P}}$-derivation) if it has been derived from $\mathcal{P}$ using instructions $\widehat{\mathbf{I}}.1$–$\widehat{\mathbf{I}}.8$ (respectively, $\widehat{\mathbf{I}}.1$–$\widehat{\mathbf{I}}.6$). Note that the instructions $\widehat{\mathbf{I}}.7$–$\widehat{\mathbf{I}}.8$ express the implication of minimal logic, and, as the reader can see, are reformulations of $\widehat{\mathbf{I}}.1$ and $\widehat{\mathbf{I}}.2$, given for the constant $\perp$ instead of an atomic formula $A$. The search instruction $\widehat{\mathbf{I}}.9$ represents intuitionistic negation.

In fact, these operational instructions correspond to inference schemas of a sequent calculus. Consider, for example, a familiar sequent calculus with antecedents and succedents being sets of first-order formulas [9]:

$$\frac{\Gamma \vdash \Delta, B \quad \Gamma \vdash \Delta, C}{\Gamma \vdash \Delta, B \wedge C} \wedge_r \qquad \frac{B, C, \Delta \vdash \Theta}{B \wedge C, \Delta \vdash \Theta} \wedge_l$$

$$\frac{\Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, B \vee C} \vee_r \qquad \frac{\Gamma \vdash \Delta, C}{\Gamma \vdash \Delta, B \vee C} \vee_r$$

$$\frac{B, \Delta \vdash \Theta \quad C, \Delta \vdash \Theta}{B \vee C, \Delta \vdash \Theta} \vee_l$$

$$\frac{\Gamma \vdash \Theta, B \quad C, \Gamma \vdash \Delta}{B \supset C, \Gamma \vdash \Delta \cup \Theta} \supset_l \qquad \frac{B, \Gamma \vdash \Theta, C}{\Gamma \vdash \Theta, B \supset C} \supset_r$$

$$\frac{B\{x/t\}, \Gamma \vdash \Theta}{\forall x B, \Gamma \vdash \Theta} \forall_l \qquad \frac{\Gamma \vdash \Theta, B\{x/t\}}{\Gamma \vdash \Theta, \exists x B} \exists_r$$

$$\frac{B\{x/c\}, \Gamma \vdash \Theta}{\exists x B, \Gamma \vdash \Theta} \exists_l \qquad \frac{\Gamma \vdash \Theta, B\{x/c\}}{\Gamma \vdash \Theta, \forall x B} \forall_r$$

$$\frac{\Gamma \vdash \Theta, \perp}{\Gamma \vdash \Theta, B} \perp_r$$

A proof for a sequent $\Gamma \vdash \Theta$ is a finite tree constructed using these inference schemas and such that the root is labelled with $\Gamma \vdash \Theta$ and the leaves are labelled with initial sequents, or axioms, i.e., sequents $\Gamma \vdash \Theta$ such that the intersection $\Gamma \cap \Theta$ contains either $\perp$ or an atomic formula.

An arbitrary proof with axioms of the kind

$$B, \Gamma \vdash \Theta, B \quad \text{or} \quad \bot, \Gamma \vdash \Theta, \bot$$

is called a **C**-proof. A **C**-proof in which each sequent occurrence has a singleton set succedent is called an **I**-proof. If an **I**-proof contains no instance of the schema $\bot_r$, it is called an **M**-proof. An **M**-proof which contains no occurrence of the constant $\bot$ is called a **P**-proof. In our notation the expressions **C**-proof, **I**-proof, **M**-proof and **P**-proof stand respectively for proofs in classical, intuitionistic, minimal and positive logics.

Note that no structural rules are mentioned here, because we treat formulas appearing in the sequents as sets, and not as sequences. This makes rules of interchange and contraction inessential for uniform proofs. As for the weakening rule, the following lemma from [5] establishes an analog of weakening in antecedent.

**Lemma 1.** Let $\Xi$ be a proof of $\Gamma \vdash \Theta$ and let $\Gamma'$ be a set of formulas. Let $\Xi + \Gamma'$ be the tree of sequents obtained by adding $\Gamma'$ to the antecedent of all sequents in $\Xi$. Then $\Xi + \Gamma'$ is a proof for $\Gamma \cup \Gamma' \vdash \Theta$.

Now we can relate the system of search instructions $\widehat{\mathbf{I}}$ and **I**-proofs: there is a strict correspondence between rules $\widehat{\mathbf{I}}$.3 and $\vee_r$, $\widehat{\mathbf{I}}$.4 and $\wedge_r$, $\widehat{\mathbf{I}}$.5 and $\exists_r$, $\widehat{\mathbf{I}}$.6 and $\supset_r$, $\widehat{\mathbf{I}}$.9 and $\bot_r$. Rules $\wedge_l$ and $\forall_l$ correspond to the use of $[P]$ instead of $P$ in the proof rules $\widehat{\mathbf{I}}$.1 and $\widehat{\mathbf{I}}$.2. However, the match between inference figures $\widehat{\mathbf{I}}$.2 and $\supset_l$ is not direct. Given the sequent $G' \supset A, P \vdash G$, $\widehat{\mathbf{I}}$.2 is applicable only if $G = A$ while $\supset_l$ has no such restriction. In fact, $\supset_l$ generates an entire subproof for $A, P \vdash G$ which is not present in instances of search instruction $\widehat{\mathbf{I}}$.2. As a result of this difference, Miller needed to introduce the following definitions and a lemma which we reproduce as Lemma 2 below.

Instances of $\supset_l$ in a proof are at the root of two smaller proofs. These two proofs are called the *left subproof* and the *right subproof* of this instance of $\supset_l$. An instance of $\supset_l$ in a proof is *simple* if its right subproof has height 1. Otherwise, the instance is *complex.* A proof in which all instances of $\supset_l$ are simple is a *simple proof.* It is the simple instances of $\supset_l$ which correspond to uses of inference figure $\widehat{\mathbf{I}}$.2.

**Lemma 2.** Let $\mathcal{P}$ be a set of definite clauses and let $G$ be a goal formula. If the sequent $\mathcal{P} \vdash G$ has an **I**-proof then it has a simple **I**-proof.

The following lemma and theorems presented in this section were proved in [5]. We state these here to show how the argument is being developed.

**Lemma 3.** A sequent has an **I**-proof with no occurrence of $\bot$ iff it has a **P**-proof.

**Theorem 4.** A goal $G$ has an $\widehat{\mathbf{M}}$-derivation from a program $\mathcal{P}$ iff the sequent $\mathcal{P} \vdash G$ has an **M**-proof.

**Theorem 5.** A goal $G$ has an $\widehat{\mathbf{I}}$-derivation from a program $\mathcal{P}$ iff the sequent $\mathcal{P} \vdash G$ has an **I**-proof.

## 3 Further extensions of operational semantics

Now we begin to investigate the question of possible extensions of the concept of a uniform proof provided that there is no change made to the language of logic programming. We present logic programming systems for reasoning in logics such as classical, relevant and paranormal.

### 3.1 Classical sequent calculus and operational semantics

In this subsection we introduce the system of search instructions which corresponds to the sequent calculus for classical logic. As a starting point, we drop the requirement that goals should be represented by single formulas and allow goals to be sets of formulas. We will denote a single goal by $G$ and sets of goals by $\mathcal{G}$. Consider the following instructions:

$\widehat{\mathbf{C}}$.1 $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{A\}$ if $A \in [\mathcal{P}]$;

$\widehat{\mathbf{C}}$.2 $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{A\}$ if there is a formula $(G \supset A) \in [\mathcal{P}]$ and $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G\}$;

$\widehat{\mathbf{C}}$.3 $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G_1 \vee G_2\}$ if $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G_1\}$ or $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G_2\}$;

$\widehat{\mathbf{C}}$.4 $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G_1 \wedge G_2\}$ if $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G_1\}$ and $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G_2\}$;

$\widehat{\mathbf{C}}$.5 $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{\exists x G\}$ if there is some term $t$ such that $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G\{x/t\}\}$;

$\widehat{\mathbf{C}}$.6 $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{D \supset G\}$ if $\mathcal{P} \cup \{D\} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G\}$;

$\widehat{\mathbf{C}}$.7 $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{\bot\}$ if, $\bot \in [P]$;

$\widehat{\mathbf{C}}$.8 $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{\bot\}$ if there is a formula $(G \supset \bot) \in [\mathcal{P}]$ and $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G\}$;

$\widehat{\mathbf{C}}$.9 $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{\bot\}$ if $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G\}$;

$\widehat{\mathbf{C}}$.10 $\mathcal{P} \cup \{G \supset \bot\} \vdash_{\widehat{\mathbf{C}}} \mathcal{G}$ if $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{G\}$;

$\widehat{\mathbf{C}}$.11 $\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{D \supset \bot\}$ if $\mathcal{P} \cup \{D\} \vdash_{\widehat{\mathbf{C}}} \mathcal{G}$.

Allowing for some loose notation, we say that a goal $G$ (or a set of goals $\mathcal{G}$) has a $\widehat{\mathbf{C}}$-derivation from a program $\mathcal{P}$ if $G$ (respectively, $\mathcal{G}$) has been derived from $\mathcal{P}$ using instructions $\widehat{\mathbf{C}}$.1–$\widehat{\mathbf{C}}$.11. Note that the instructions $\widehat{\mathbf{C}}$.1–$\widehat{\mathbf{C}}$.9 are precisely instructions $\widehat{\mathbf{I}}$.1–$\widehat{\mathbf{I}}$.9, which are modified to allow for multiple goals. Instructions $\widehat{\mathbf{C}}$.10–$\widehat{\mathbf{C}}$.11 introduce classical negation.

- The search instruction $\widehat{\mathbf{C}}$.10 would work as follows. Suppose we need to prove that a set of goals $\mathcal{G}$ can be derived from a program $\mathcal{P} \cup \{G \supset \bot\}$. Then either $G \supset \bot$ does not influence the inference of $\mathcal{G}$, and then $\mathcal{G}$ can be derived from $\mathcal{P}$; or, if we cannot deduce $\mathcal{G}$ from $\mathcal{P}$, we conclude that $G \supset \perp$ influences the proof of $\mathcal{G}$ and now it suffices to show that $G$ can be derived from $\mathcal{P}$. Then if we add $G \supset \perp$ to the program, we will get a contradiction and from this contradiction we will be able to derive any set of goals, in particular, the desired set $\mathcal{G}$. Thus, instruction $\widehat{\mathbf{C}}.10$ subsumes instruction $\widehat{\mathbf{C}}.9$.

- The search instruction $\widehat{\mathbf{C}}.11$ says that the negation of a formula $D$ can be derived from $\mathcal{P}$ if, being added to $\mathcal{P}$, this formula does not generate new consequences of $\mathcal{P}$.

These two instructions would correspond to the following rules in classical sequent calculus:

$$\frac{\Gamma \vdash \Theta, B}{B \supset \perp, \Gamma \vdash \Theta}\neg_l \qquad \frac{B, \Gamma \vdash \Theta}{\Gamma \vdash \Theta, B \supset \perp}\neg_r \ .$$

Axioms of $\widehat{\mathbf{C}}$ are respectively

$$\mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{A\}, \text{ where } A \in [\mathcal{P}] \quad \text{ or } \quad \mathcal{P} \vdash_{\widehat{\mathbf{C}}} \mathcal{G} \cup \{\perp\}, \text{ where } \perp \in [\mathcal{P}] \ .$$

We can show that the rule $\neg_l$ can in fact substitute the rule $\perp_r$.

$$\dfrac{\dfrac{\perp \vdash \perp}{\vdash \perp \supset \perp}\supset_l \qquad \dfrac{\dfrac{\dfrac{\Xi}{\Gamma \vdash \Theta, \perp}}{\Gamma \vdash \Theta, \perp, B}\text{weakening}}{\perp \supset \perp, \Gamma \vdash \Theta, B}\neg_l}{\Gamma \vdash \Theta, B}\text{cut}$$

This suggests that the instruction $\widehat{\mathbf{C}}.9$ can be derived using instruction $\widehat{\mathbf{C}}.10$.

**Lemma 6.** If a goal $G$ has an $\widehat{\mathbf{I}}$-derivation from $\mathcal{P}$, it has a $\widehat{\mathbf{C}}$-derivation from $\mathcal{P}$, but not converse.

To prove that the converse part of Lemma 6 does not hold, it is sufficient to show that a goal $A_1$ has a $\widehat{\mathbf{C}}$-derivation from the program $\{(A_1 \supset A_2) \supset A_1\}$, but does not have any $\widehat{\mathbf{I}}$-derivations, because only single formulas are allowed to appear in the goals of $\widehat{\mathbf{I}}$-instructions.

The following two lemmas will be useful when allowing more then one formula in succedent. First one is the analog of the succedent weakening.

**Lemma 7.** Let $\Xi$ be a proof of $\Gamma \vdash \Theta$ and let $\Theta'$ be a set of formulas. Let $\Xi + \Theta'$ be the tree of sequents obtained by adding $\Theta'$ to the succedent of all sequents in $\Xi$. Then $\Xi + \Theta'$ is a proof for $\Gamma \vdash \Theta \cup \Theta'$.

Now the instruction $\widehat{\mathbf{C}}.11$ can be derived from the rule $\widehat{\mathbf{C}}.6$ and the lemma 7, namely from the fact that in a derivation of any goal (or set of goals) $\perp$ can be added to the initial program through all the steps of inference. This is why from now on we will think of $\widehat{\mathbf{C}}$ as being defined by search instructions $\widehat{\mathbf{C}}.1$–$\widehat{\mathbf{C}}.10$. And $\widehat{\mathbf{C}}$-derivation will stand for the derivation which uses instructions $\widehat{\mathbf{C}}.1$–$\widehat{\mathbf{C}}.10$.

Analogously to the case of operational inference system for intuitionistic logic, in the case of classical logic we do not introduce a search instruction corresponding to $\supset_l$ of classical sequent calculus. Instead, we make use of the reduction of **C**-proofs to simple **C**-proofs. For this purpose the following extension of Lemma 11 from [5] to **C**-proofs suffices.

**Lemma 8.** Assume that $\Xi$ is a **C**-proof of the form

$$\dfrac{\begin{array}{c}\Xi_1\\ \mathcal{P} \vdash \mathcal{G}_1, G'\end{array} \qquad \begin{array}{c}\Xi_2\\ A, \mathcal{P} \vdash \mathcal{G}_2, G\end{array}}{G' \supset A, \mathcal{P} \vdash \mathcal{G}_1 \cup \mathcal{G}_2, G} \supset_l$$

where $\Xi_1$ and $\Xi_2$ are simple **C**-proofs. Then there exists a simple **C**-proof for

$$G' \supset A, \mathcal{P} \vdash \mathcal{G}_1 \cup \mathcal{G}_2, G \ . \tag{1}$$

Additionally, if $A$ is replaced with $\perp$ in the above proof tree then there exists a simple **C**-proof for

$$G' \supset \perp, \mathcal{P} \vdash \mathcal{G}_1 \cup \mathcal{G}_2, G \ . \tag{2}$$

*Proof.* The proof proceeds by induction on the height of $\Xi_2$. If the height is 1 then $\Xi_2$ consists only of its root sequent, the right premise of the instance of $\supset_l$, and therefore this instance of $\supset_l$ is simple. For the inductive step, assume that the height of $\Xi_2$ is $n > 1$. To complete the proof it suffices to show that this complex instance of $\supset_l$ commutes with the last inference schema instance in $\Xi_2$, or, in other words, to show that the instance of sequent (1) (or (2), for the case of $\perp$) has another **C**-proof in which all the instances of the schema $\supset_l$ have their right **C**-subproofs being of height less than $n$ (and hence, the root sequents of these instances would yield simple **C**-proofs by the induction hypothesis).

Consider, for instance, the case $\wedge_r$. Assume $\Xi$ is a proof of the following form.

$$\dfrac{\begin{array}{c}\Xi_1\\ \mathcal{P} \vdash \mathcal{G}_1, G'\end{array} \qquad \dfrac{\begin{array}{c}\Xi_2'\\ A, \mathcal{P} \vdash \mathcal{G}_2, G_1\end{array} \qquad \begin{array}{c}\Xi_2''\\ A, \mathcal{P} \vdash \mathcal{G}_2, G_2\end{array}}{A, \mathcal{P} \vdash \mathcal{G}_2, G_1 \wedge G_2} \wedge_r}{G' \supset A, \mathcal{P} \vdash \mathcal{G}_1 \cup \mathcal{G}_2, G_1 \wedge G_2} \supset_l$$

In this case $\supset_l$ perfectly commutes with $\wedge_r$ as shown below.

$$\dfrac{\dfrac{\begin{array}{c}\Xi_1\\ \mathcal{P} \vdash \mathcal{G}_1, G'\end{array} \qquad \begin{array}{c}\Xi_2'\\ A, \mathcal{P} \vdash \mathcal{G}_2, G_1\end{array}}{G' \supset A, \mathcal{P} \vdash \mathcal{G}_1 \cup \mathcal{G}_2, G_1} \supset_l \qquad \dfrac{\begin{array}{c}\Xi_1\\ \mathcal{P} \vdash \mathcal{G}_1, G'\end{array} \qquad \begin{array}{c}\Xi_2''\\ A, \mathcal{P} \vdash \mathcal{G}_2, G_2\end{array}}{G' \supset A, \mathcal{P} \vdash \mathcal{G}_1 \cup \mathcal{G}_2, G_2} \supset_l}{G' \supset A, \mathcal{P} \vdash \mathcal{G}_1 \cup \mathcal{G}_2, G_1 \wedge G_2} \wedge_r$$

Obviously, this commutativity also holds if $\bot$ has been taken instead of $A$.

The cases $\supset_r$, $\vee_r$, $\exists_r$, $\forall_l$ and $\wedge_l$ are obtainable using the same method. These cases are also straightforward generalizations of those from Lemma 11 in [5].

Consider the case $\supset_l$. Assume $\Xi$ is the following proof tree.

$$\dfrac{\begin{array}{c}\Xi_1\\ G_1 \supset A_1, \mathcal{P}' \vdash \mathcal{G}_1, G'\end{array} \qquad \dfrac{\begin{array}{c}\Xi_2'\\ A, \mathcal{P}' \vdash \mathcal{G}_2', G_1\end{array} \qquad A, A_1, \mathcal{P}' \vdash \mathcal{G}_2'', G}{A, G_1 \supset A_1, \mathcal{P}' \vdash \mathcal{G}_2' \cup \mathcal{G}_2'', G}\supset_l}{G' \supset A, G_1 \supset A_1, \mathcal{P}' \vdash \mathcal{G}_1 \cup \mathcal{G}_2' \cup \mathcal{G}_2'', G}\supset_l$$

The topmost right sequent is initial; hence, either (a) $\mathcal{P}' \cap (\{G\} \cup \mathcal{G}_2'') \neq \emptyset$, or (b) $A \in \{G\} \cup \mathcal{G}_2''$, or (c) $A_1 \in \{G\} \cup \mathcal{G}_2''$. In the case (a) the root sequent of $\Xi$ is an instance of an axiom. For (b), the simple **C**-proof has the form

$$\dfrac{\begin{array}{c}\Xi_1\\ G_1 \supset A_1, \mathcal{P}' \vdash \mathcal{G}_1, G'\end{array} \qquad A, G_1 \supset A_1, \mathcal{P}' \vdash \mathcal{G}_2' \cup \mathcal{G}_2'', G}{G' \supset A, G_1 \supset A_1, \mathcal{P}' \vdash \mathcal{G}_1 \cup \mathcal{G}_2' \cup \mathcal{G}_2'', G}\supset_l$$

and, for (c), the corresponding simple **C**-proof is of the kind

$$\supset_l \dfrac{\dfrac{\begin{array}{c}\Xi_1\\ G_1 \supset A_1, \mathcal{P}' \vdash \mathcal{G}_1, G'\end{array} \qquad \begin{array}{c}\Xi_2' + \{G_1 \supset A_1\}\\ A, G_1 \supset A_1, \mathcal{P}' \vdash \mathcal{G}_2', G_1\end{array}}{G' \supset A, G_1 \supset A_1, \mathcal{P}' \vdash \mathcal{G}_1 \cup \mathcal{G}_2', G_1}\supset_l \qquad A_1, G' \supset A, \mathcal{P}' \vdash \mathcal{G}_2'', G}{G' \supset A, G_1 \supset A_1, \mathcal{P}' \vdash \mathcal{G}_1 \cup \mathcal{G}_2' \cup \mathcal{G}_2'', G}$$

where $\Xi_2' + \{G_1 \supset A_1\}$ is shorter than $\Xi_2$ and, by the inductive hypothesis, has a simple **C**-proof.

The last two cases $\neg_l$ and $\neg_r$ are straightforward since $\supset_l$ commutes with $\neg_l$ and with $\neg_r$ without any problems. □

**Lemma 9.** If $\mathcal{P} \vdash \mathcal{G}$ has a **C**-proof then it has a simple **C**-proof.

*Proof.* Follows by induction on the number of complex instances of $\supset_l$ in a **C**-proof of $\mathcal{P} \vdash \mathcal{G}$. According to Lemma 8, the complex instances which have only simple left and right subproofs can be converted into simple ones. Hence, all the complex instances can be removed. □

**Theorem 10.** A set of goals $\mathcal{G}$ has a $\widehat{\mathbf{C}}$-derivation from a program $\mathcal{P}$ iff the sequent $\mathcal{P} \vdash \mathcal{G}$ has a **C**-proof.

To prove this theorem it is only needed to show how, given a **C**-proof $\Xi$ for $\mathcal{P} \vdash \mathcal{G}$, we can obtain a $\widehat{\mathbf{C}}$-derivation for $\mathcal{G}$ from $\mathcal{P}$. This can be done by induction on the height of $\Xi$ using Lemma 9 and considering all the possible inference schemas in the root of $\Xi$ during the inductive step.

We have shown that proof-theoretic operational inference systems can be extended to the operational system which corresponds to classical sequent calculus. This extension has its price. In $\widehat{\mathbf{C}}$-instructions we employed sets of goals (that is disjunctions of goals). This leads an operational system answering to particular goal formulas in queries (sets of goal formulas) without making any references to these particular formulas. It shall be unclear in general the variables in which goal formulas should be substituted by their values.

### 3.2 Towards relevance logic

It is widely known that the system **R** of relevance logic is undecidable even at the propositional level [2]. However, there exists a sequent calculus for **R** defined also in [2]. This fact at least provides us with a possibility to relate relevance logic and operational inference systems. Unfortunately, even a brief look at the problem of relating them shows the impossibility of developing an operational inference system for full **R** in a way similar to the case of classical logic. This happens because **R** has no natural sequent calculus with antecedent and succedent as sets. In fact, **R** has a sequent calculus where succedent is a sequence and antecedent is a sequence of sequences of sequences! Therefore, let us consider a suitable fragment of **R**, namely **LR**, the so-called "lattice **R**". First of all, note that propositional **LR** is decidable (with very bad complexity though, as is customary for logics of relevance), and its sequent calculus enjoys cut elimination.

Using the translation of **LR** into the implication-conjunction fragment of **R** given by Meyer (further references are found in [2], Section 4.8), we define the sequent calculus for **LR** with antecedents and succedents as sets of formulas. Taking antecedents and succedents to be sets we lose the possibility to express multiplicities of formulas in sequents, and, hence, the resulting sequent calculus is less expressive than sequent calculus for **LR** with sequences. Fortunately, for the purpose of developing an operational inference system we do not need such expressiveness. To reason about formal sequents of the kind $\mathcal{P} \vdash \mathcal{G}$, where $\mathcal{P}$ is a logic program, and $\mathcal{G}$ is a set of goals, we only need some sequent calculus with antecedents and succedents being sets of formulas (to check this, simply observe that both $\mathcal{P}$ and $\mathcal{G}$ are sets).

Consider now the set of inference schemas consisting of $\wedge_r$, $\vee_r$, $\vee_l$, $\forall_l$, $\exists_r$, $\forall_r$ and $\exists_l$ from Section 2, and schemas for negation $\neg_l$ and $\neg_r$ from Subsection 3.1 along with new schemas for conjunction:

$$\frac{B, \Delta \vdash \Theta}{B \wedge C, \Delta \vdash \Theta}\ \wedge_l^{\mathbf{LR}} \qquad \frac{C, \Delta \vdash \Theta}{B \wedge C, \Delta \vdash \Theta}\ \wedge_l^{\mathbf{LR}}$$

and new schemas for implication:

$$\frac{\Gamma \vdash \Theta, B \quad C, \Gamma \vdash \Delta}{B \supset C, \Gamma \vdash \Delta \cup \Theta}\ \supset_l^{\mathbf{LR}} \qquad \frac{B, \Gamma \vdash \Theta, C}{\Gamma \vdash \Theta, B \supset C}\ \supset_r^{\mathbf{LR}}$$

where in both $\supset_l^{\mathbf{LR}}$ and $\supset_r^{\mathbf{LR}}$ $C$ is not allowed to be $\bot$. A proof is to be called an $\mathbf{LR}^{set}$-*proof* if it consists of instances of inference schemas listed in this paragraph, provided that the only axioms of that proof are of the form

$$B \vdash B \quad \text{(including the case } \bot \vdash \bot) \ .$$

Restricting axioms to just one-element ones we take control over the structural rule of weakening which (or, better said, the unavailability of which) is crucial for relevance logic, and for **LR** in particular. In other words, this restriction of axioms in our calculus with antecedents and succedents as sets corresponds to rejecting the weakening rule in a traditional calculus with antecedents and succedents as sequences.

Let $\mathcal{P}$ be a logic program. Define $\lceil \mathcal{P} \rceil$ be the smallest set of formulas satisfying the following recursive conditions:

- $\mathcal{P} \subseteq \lceil \mathcal{P} \rceil$;
- if $D_1 \wedge D_2 \in \lceil \mathcal{P} \rceil$ then $D_1 \in \lceil \mathcal{P} \rceil$ *or* $D_2 \in \lceil \mathcal{P} \rceil$;
- if $\forall x D \in \lceil \mathcal{P} \rceil$ then $D\{x/t\} \in \lceil \mathcal{P} \rceil$ for *some* term $t$.

Thus defined $\lceil \mathcal{P} \rceil$ might be called the *non-deterministic closure of* $\mathcal{P}$. The operation $\lceil \cdot \rceil$ has prominent non-deterministic features which result from a specific approach to conjunction and universal quantifier.

Now we are in a position to describe an operational inference system $\widehat{\mathbf{LR}}$ for logic programming language of lattice relevance logic. Let $\mathcal{P}$, $\mathcal{G}$, $A$, $D$, $G$ and $G_i$ be as above. The set of search instructions for $\widehat{\mathbf{LR}}$ is the following:

$\widehat{\mathbf{LR}}$.1 $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \{A\}$ if, for some $\overline{\mathcal{P}} = \lceil \mathcal{P} \rceil$, $A$ is the only atomic formula in $\overline{\mathcal{P}}$;

$\widehat{\mathbf{LR}}$.2 $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{A\}$ if there is a formula $(G \supset A) \in \lceil \mathcal{P} \rceil$ and $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{G\}$;

$\widehat{\mathbf{LR}}$.3 $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{G_1 \vee G_2\}$ if $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{G_1\}$ or $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{G_2\}$;

$\widehat{\mathbf{LR}}$.4 $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{G_1 \wedge G_2\}$ if $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{G_1\}$ and $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{G_2\}$;

$\widehat{\mathbf{LR}}$.5 $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{\exists x G\}$ if there is some term $t$ such that $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{G\{x/t\}\}$;

$\widehat{\mathbf{LR}}$.6 $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{D \supset G\}$ if $\mathcal{P} \cup \{D\} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{G\}$, provided that $G \neq \bot$;

$\widehat{\mathbf{LR}}$.7 $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \{\bot\}$ if, for some $\overline{\mathcal{P}} = \lceil \mathcal{P} \rceil$, $\bot$ is the only atomic formula in $\overline{\mathcal{P}}$;

$\widehat{\mathbf{LR}}$.8 $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{\bot\}$ if there is a formula $(G \supset \bot) \in \lceil \mathcal{P} \rceil$ and $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{G\}$;

$\widehat{\mathbf{LR}}.9$ $\quad \mathcal{P} \cup \{G \supset \bot\} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G}$ if $\mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{G\}$;

$\widehat{\mathbf{LR}}.10$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G} \cup \{D \supset \bot\}$ if $\mathcal{P} \cup \{D\} \vdash_{\widehat{\mathbf{LR}}} \mathcal{G}$.

The results analogous to Lemmas 8 and 9 hold in the case of $\mathbf{LR}^{set}$-proofs. Therefore, Theorem 11 is provable analogously to Theorem 10.

**Theorem 11.** A set of goals $\mathcal{G}$ has an $\widehat{\mathbf{LR}}$-derivation from a program $\mathcal{P}$ iff the sequent $\mathcal{P} \vdash \mathcal{G}$ has an $\mathbf{LR}^{set}$-proof.

This theorem also justifies the connection between the operational system $\widehat{\mathbf{LR}}$ and the sequent calculus for lattice relevance logic **LR** as defined in [2]. The connection is precisely in that a formal sequent $\{D_1, \ldots, D_m\} \vdash \{G_1, \ldots, G_n\}$ with sets can be translated into a sequent $D_1, \ldots, D_m \vdash G_1, \ldots, G_n$ with sequences; and then the set of goals $\{G_1, \ldots, G_n\}$ is $\widehat{\mathbf{LR}}$-derivable from the logic program $\{D_1, \ldots, D_m\}$ if and only if the sequent $\{D_1, \ldots, D_m\} \vdash \{G_1, \ldots, G_n\}$ with sets has an $\mathbf{LR}^{set}$-proof if and only if the sequent $D_1, \ldots, D_m \vdash G_1, \ldots, G_n$ with sequences has an **LR**-proof.

### 3.3 Paranormal logic

Consider the following definition. Given a first-order language $L$, assume we have defined a sequent-style proof procedure for formulas determined by $L$. A set of all formulas of $L$ which can be obtained from initial sequents using a given set of rules will be called a *theory* (denote it by $T$). If there is a formula $B$ such that $B \in T$ and $(B \supset \bot) \in T$, the theory $T$ is called *inconsistent*. If a theory is inconsistent but is not equal to the set of all formulas over $L$, it is called *paraconsistent*. If, for each formula $B$, either $B \in T$ or $(B \supset \bot) \in T$ then $T$ is called *complete*. If a theory $T$ is incomplete but every theory $T'$ containing $T$ is complete, $T$ is called *paracomplete*. A theory which is both paraconsistent and paracomplete is called *paranormal*.

We refer to [7] and less firmly to [6] for definition of a sequent calculus for paranormal logic. In fact, the rules $\wedge_r$, $\wedge_l$, $\vee_r$, $\vee_l$, $\supset_r$, $\supset_l$, $\forall_r$, $\forall_l$, $\exists_r$, $\exists_l$ are precisely those which were defined in Section 2. The structural rules are interchange, contraction, weakening and cut, which can be eliminated. The only difference is in rules for negation, and they are as follows:

$$\frac{\Gamma \vdash \Theta, B}{B \supset \bot, \Gamma \vdash \Theta}\,\neg_l \qquad \frac{B, \Gamma \vdash \Theta}{\Gamma \vdash \Theta, B \supset \bot}\,\neg_r$$

where $B$ is not atomic, and

$$\frac{\Gamma \vdash \Theta, B}{\Gamma \vdash \Theta, (B \supset \bot) \supset \bot}\,\neg\neg_l \qquad \frac{B, \Gamma \vdash \Theta}{(B \supset \bot) \supset \bot, \Gamma \vdash \Theta}\,\neg\neg_r$$

where $B$ is atomic.

Moreover, we can differentiate between these rules and assert that the restriction of $\neg_l$ (respectively, $\neg_r$) gives us the sequent calculus corresponding to paraconsistent (respectively, paracomplete logic). We will denote paranormal proofs as **PN**-proofs.

Now, using $\widehat{\mathbf{C}}$-instructions, we can introduce the search instructions which will correspond to the paranormal sequent calculus. We have shown in Subsection 3.1 that rules $\neg_r$ and $\neg_l$ correspond to the instructions $\widehat{\mathbf{C}}.10$ and $\widehat{\mathbf{C}}.11$. Now we can manipulate these instructions in order to relate them to paranormal sequent calculus. The instruction $\widehat{\mathbf{C}}.11$ was shown to be derivable from the instructions $\widehat{\mathbf{C}}.6$ and a form of weakening rule, and we eliminated it from the $\widehat{\mathbf{C}}$-derivations. This is why it is impossible to introduce any changes in the rule $\widehat{\mathbf{C}}.11$. However, we can restrict the rule $\widehat{\mathbf{C}}.6$ for implicative formulas containing $\perp$ in the consequent of implication and thus we will establish the desired connection with paranormal logic. We introduce the following search instructions:

$\widehat{\mathbf{PN}}.1$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{A\}$ if $A \in [\mathcal{P}]$;

$\widehat{\mathbf{PN}}.2$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{A\}$ if there is a formula $(G \supset A) \in [\mathcal{P}]$ and $\mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G\}$;

$\widehat{\mathbf{PN}}.3$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G_1 \vee G_2\}$ if $\mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G_1\}$ or $\mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G_2\}$;

$\widehat{\mathbf{PN}}.4$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G_1 \wedge G_2\}$ if $\mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G_1\}$ and $\mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G_2\}$;

$\widehat{\mathbf{PN}}.5$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{\exists x G\}$ if there is some term $t$ such that $\mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G\{x/t\}\}$;

$\widehat{\mathbf{PN}}.6$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{D \supset G\}$ if $\mathcal{P} \cup \{D\} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G\}$, where $D$ is not atomic whenever $G = \perp$;

$\widehat{\mathbf{PN}}.7$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{\perp\}$ if, $\perp \in [P]$;

$\widehat{\mathbf{PN}}.8$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{\perp\}$ if there is a formula $(G \supset \perp) \in [\mathcal{P}]$ and $\mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G\}$;

$\widehat{\mathbf{PN}}.9$ $\quad \mathcal{P} \cup \{G \supset \perp\} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G}$ if $\mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G\}$, where $G$ is not atomic;

$\widehat{\mathbf{PN}}.10$ $\quad \mathcal{P} \cup \{D\} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G}$ if $\mathcal{P} \cup \{(D \supset \perp) \supset \perp\} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G}$, where $D$ is atomic.

$\widehat{\mathbf{PN}}.11$ $\quad \mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{G\}$ if $\mathcal{P} \vdash_{\widehat{\mathbf{PN}}} \mathcal{G} \cup \{(G \supset \perp) \supset \perp\}$, where $G$ is atomic.

We say that a goal $G$ (or a set of goals $\mathcal{G}$) has a $\widehat{\mathbf{PN}}$-derivation from a program $\mathcal{P}$ if $G$ (respectively, $\mathcal{G}$) has been derived from $P$ using the search instructions $\widehat{\mathbf{PN}}.1$–$\widehat{\mathbf{PN}}.11$. Note that the instructions $\widehat{\mathbf{PN}}.10$–$\widehat{\mathbf{PN}}.11$ compensate some restrictions made in $\widehat{\mathbf{PN}}.6$ and $\widehat{\mathbf{PN}}.9$, and thus allow us to derive, for example, the goal $A$ from a program $\{(A \supset \perp) \supset \perp\}$, or the goal $(A \supset \perp) \supset \perp$ from a program $\{A\}$.

**Lemma 12.** If a goal $G$ has a $\widehat{\mathbf{PN}}$-derivation from $\mathcal{P}$, it has a $\widehat{\mathbf{C}}$-derivation from $\mathcal{P}$, but not converse.

The converse of this lemma can be disproved if we consider the program $\{A_1 \supset \bot\}$ and the goal $A_1 \supset A_2$, where $A_1$ is an atom. This goal has $\widehat{\mathbf{C}}$-derivation, but has no $\widehat{\mathbf{PN}}$-derivation from the program.

Then the following theorem can be established.

**Theorem 13.** A set of goals $\mathcal{G}$ has a $\widehat{\mathbf{PN}}$-derivation from a program $\mathcal{P}$ iff the sequent $\mathcal{P} \vdash \mathcal{G}$ has a $\mathbf{PN}$-proof.

The proof is based on Theorem 10. We only need to show the correspondence between search instructions $\widehat{\mathbf{PN}}.6$, $\widehat{\mathbf{PN}}.9$, $\widehat{\mathbf{PN}}.10$, $\widehat{\mathbf{PN}}.11$ and, respectively, restricted rules for $\neg_l$, $\neg_r$, $\neg\neg_l$, $\neg\neg_r$, which is straightforward. There are some consequences of Theorem 13 which need to be mentioned.

The paraconsistent operational inference system can work with inconsistencies without making a set of derivable formulas trivial. The paracomplete operational semantics is capable of handling the possible incompleteness of a given logic program in case when we need to work with proofs which do not guarantee for each formula $B$ a derivation for either $B$ or $\neg B$. These are the nice properties of paranormal logics which can be useful when one wishes to employ logic programs to manage his inconsistent and/or incomplete databases.

The second consequence concerns the possible relations of the operational semantics to intuitionistic paranormal logics. The reader can easily check that the following formulas have $\widehat{\mathbf{PN}}$-derivation: $B \vee (B \supset \bot)$, if $B$ is not atomic and $((B \supset \bot) \supset \bot) \supset B$ if $B$ is arbitrary. Moreover, $\widehat{\mathbf{PN}}$-proofs along with $\widehat{\mathbf{C}}$-proofs have the non-deterministic property. All this motivates us to look for another type of paraconsistent operational inference system. In [8] was defined a paranormal logic which is intuitionistically acceptable (we will abbreviate it **IPN**). The **IPN**-proof is a **PN**-proof whose sequents have only single formulas in succedents and no instances of the rule $\neg\neg_l$ occur in the proof. We can obtain the operational semantics corresponding to this logic by restricting $\widehat{\mathbf{PN}}$ as follows: we require a set of goals in each search instruction in $\widehat{\mathbf{IPN}}$ to consist of a single formula (in the same way as in $\widehat{\mathbf{I}}$), and we eliminate the operational rule $\widehat{\mathbf{PN}}.11$. We will call this operational inference system $\widehat{\mathbf{IPN}}$. Then we can prove the following lemmas and a theorem.

**Lemma 14.** If a goal $G$ has an $\widehat{\mathbf{IPN}}$-derivation from $\mathcal{P}$, then it has a $\widehat{\mathbf{PN}}$-derivation from $\mathcal{P}$, but not converse.

To check that converse of the above lemma does not hold, consider, for example, a program $\{(B \supset \bot) \supset \bot\}$ and the goal $B$. The goal has $\widehat{\mathbf{PN}}$-derivation, but has no $\widehat{\mathbf{IPN}}$-derivation.

**Lemma 15.** If a goal $G$ has an $\widehat{\mathbf{IPN}}$-derivation from $\mathcal{P}$, then it has an $\widehat{\mathbf{I}}$-derivation from $\mathcal{P}$, but not converse.

A counterexample to the above lemma is the following. If we take a program $\{A_1 \supset \bot\}$ and a goal $A_1 \supset A_2$, where $A_1$ is an atom, the goal will have an $\widehat{\mathbf{I}}$ derivation, but will not have any $\widehat{\mathbf{IPN}}$-derivations.

The following theorem holds.

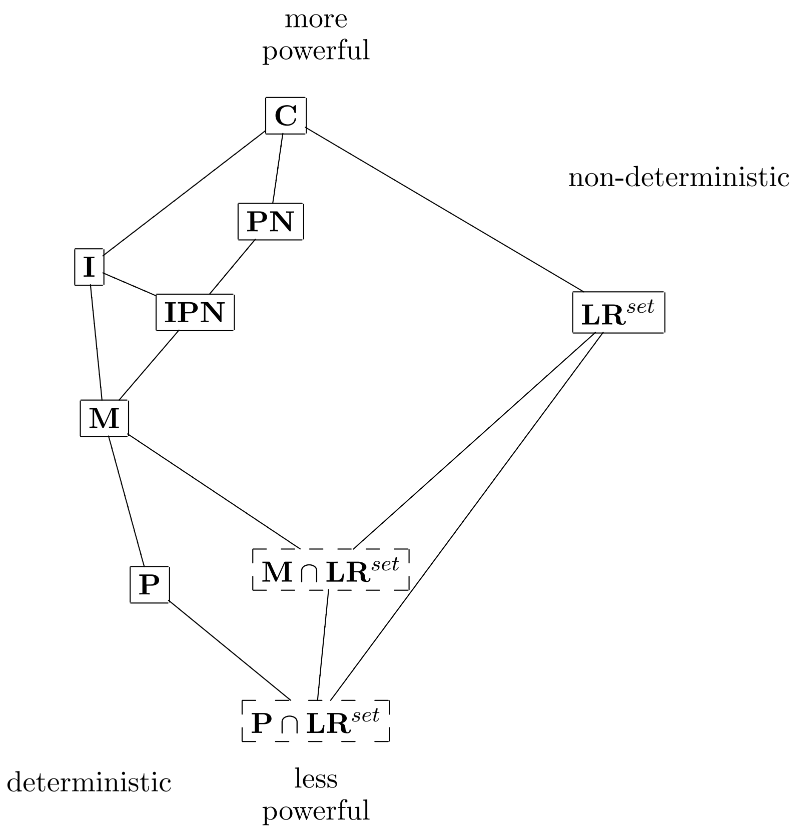
**Theorem 16.** A set of goals $\mathcal{G}$ has an $\widehat{\mathbf{IPN}}$-derivation from a program $\mathcal{P}$ iff the sequent $\mathcal{P} \vdash \mathcal{G}$ has an **IPN**-proof.

Thus we obtained operational inference system which is paraconsistent and yet intuitionistically acceptable.

# 4 Conclusions

In the paper we developed several first-order operational inference systems for logic programming in classical, lattice relevance, paranormal and intuitionistically acceptable paranormal logics. The relationships between those operational systems are schematically represented in Figure 1. The operational system for classical logic programming arose as an extension of the operational system for logic programming given by Miller in [5]. For the purposes of extension we employed the fact that the sequent calculus of classical logic was an extension of the one of intuitionistic logic. Then, treating classical logic as a universal logic (in a sense that classical logic is maximal among any other first-order logics of the same signature), we obtained other operational systems for logics which possessed sequent calculi defined by means of certain restrictions applied to sequent calculus of classical logic. Also in Figure 1 there are obvious intersections of proof systems (in dashed rectangles) which we did not study in particular.

The idea to relate logic programs and sequents allowed us, using a few simple observations, to derive operational systems straight from sequent calculi in which antecedents and succedents were sets. The derived operational systems, though clearly weaker than the original sequent calculi, suit much better for logic programming because they are designed specifically for the language of logic programming which is definitionally equivalent to the language of first-order Horn clauses. In Figure 1, the proof systems are placed according to two orderings. One orders the systems by their deductive power in the direction from bottom to top. Another orders them by determinism of their inference rules from the leftmost bottom corner to the rightmost top corner. The latter ordering is probably the most curious one since it shows that "the most philosophical" system of logical programming there, $\widehat{\mathbf{LR}}$, is the less deterministic. More precisely, already in the case of $\widehat{\mathbf{PL}}$ there is a non-determinism in that an answer (a substitution for individual




| | |
|---|---|
| $\mathbf{C}$ — classical ($\widehat{\mathbf{C}}$) | $\mathbf{PN}$ — paranormal ($\widehat{\mathbf{PN}}$) |
| $\mathbf{I}$ — intuitionistic ($\widehat{\mathbf{I}}$) | $\mathbf{IPN}$ — intuitionistic paranormal ($\widehat{\mathbf{IPN}}$) |
| $\mathbf{M}$ — minimal ($\widehat{\mathbf{M}}$) | $\mathbf{LR}^{set}$ — lattice relevance with sets ($\widehat{\mathbf{LR}}$) |
| $\mathbf{P}$ — positive ($\widehat{\mathbf{M}}$) | |

Figure 1: Relationship between the proof systems with respect to deductive power and determinism of inference rules

variables) in a logic program to a set of goals doesn't carry any information about which formula in the set of goals it is a substitution for. In $\widehat{\mathbf{LR}}$ this non-deterministic feature persists, and, moreover, another strong non-determinism appears in the closure operator.

At the extent presented here, the studies of operational inference systems for philosophical logics are in a purely theoretical phase. But once we have spoken of logic programming we can envisage some practical implementations of these operational systems. Therefore, among the possible directions for further research there are studies of different search strategies for search instructions, including assigning priorities to search instructions. And, certainly, the most interesting practical aim (and, in fact, the most practical one) is building a working interpreter for logic programming in philosophical logics.

**Acknowledgements.** The authors thank Oleg Grigoriev and Dmitri Zaitsev for a timely supply of reading materials. The first author acknowledges the partial support of the School of Mathematics, Applied Mathematics and Statistics, UCC; and the second author acknowledges the support of BCRI, UCC.

## References

[1] M. Baldoni, L. Giordano, and A. Martelli. A multimodal logic to define modules in logic programming. In D. Miller, editor, *Proc. of the International Logic Programming Symposium, ILPS'93*, pages 473–487, Vancouver, 1993. The MIT Press.

[2] J. M. Dunn and G. Restall. Relevance logic. In D. Gabbay and F. Guenther, editors, *The Handbook of Philosophical Logic*, volume 6, pages 1–136. Kluwer, 2nd edition, 2002.

[3] J. A. Harland and D. J. Pym. A uniform proof-theoretic investigation of linear logic programming. *Journal of Logic and Computation*, 4(2):175–207, 1994.

[4] J. W. Lloyd. *Foundations of Logic Programming.* Springer, Berlin, 2nd edition, 1988.

[5] D. Miller. A logical analysis of modules in logic programming. *Journal of Logic Programming*, 6(1–2):79–108, 1989.

[6] V. M. Popov. On the logics related to Arruda's system V1. *Logic and Logical Philosophy*, 7, 1999.

[7] V. M. Popov. On a four-valued paranormal logic. In *Proc. of IV All-Russian Philosophical Congress*, Moscow State University, 24–28 May 2005. In Russian.

[8] V. M. Popov and G. N. Shuklin. Intuitionistically admissible paranormal logic. In *Logicheskiye Issledovaniya*, volume 11, pages 243–246. Nauka, Moscow, 2003. In Russian.

[9] D. Prawitz. *Natural Deduction.* Almqvist & Wiksell, Uppsala, 1965.