Correct Answers for First Order Logic

Gianluca Amato
Università di Pisa, Dipartimento di Informatica
amato@di.unipi.it

Abstract

Working within a semantic framework for sequent calculi developed in [2], we propose a couple of extensions to the concepts of correct answers and correct resultants which can be applied to the full first order logic. With respect to previous proposals, this is based on proof theory rather than model theory. We motivate our choice with several examples and we show how to use correct answers to reconstruct an abstraction which is widely used in the static analysis of logic programs, namely groundness. As an example of application, we present a prototypical top-down static interpreter for properties of groundness which works for the full intuitionistic first order logic.

Keywords: first order logic, logic programming, correct answers, resultants, abstract interpretation, groundness.

1 Introduction

One of the greatest benefits of logic programming, as presented in [15], is that it is based upon the notion of *executable specifications*. The text of a logic program is endowed with both an operational (algorithmic) interpretation and an independent mathematical meaning which agree each other in several ways. The problem is that operational expressiveness (intended as the capability of directing the flow of execution of a program) tends to obscure the declarative meaning. Research in logic programming strives to find a good balance between these opposite needs.

Uniform proofs [16] have widely been accepted as one of the main tools for approaching the problem and to distinguish between logic without a clear computational flavor and logic programming languages. However, that of uniform proofs being a concept heavily based on proof theory, researches conducted along this line have always been quite far from the traditional approach based on fixpoint semantics. In turn, this latter tradition has brought up several important results concerning the effective utilization of Horn clauses as a real programming language. Among the others, problems such as compositionality of semantics [8], modularity [5, 7], static analysis [12], debugging [9], have been tackled in this setting. Adapting these results to the new logic languages developed via the proof theoretic approach, such as λProlog [18] or LinLog [3], would probably require at least two things:

- provide a fixpoint semantics for these new languages;

- generalize a great number of concepts whose definition is too much tied to the case of Horn clauses.

In [2], the authors propose a semantic framework which can be useful in such an effort. The main idea is to recognize proofs in the sequent calculi as the general counterpart of SLD resolutions for positive logic programs. Thus, the three well-known semantics (operational, declarative and fixpoint) for Horn clause logic can be reformulated within this general setting and directly applied to all the logic languages based on sequent calculi.

Classical abstractions such as correct answers or resultants, used in the semantic studies of logic programs, and abstractions for static analysis like groundness, can be retrieved in terms of properties of proofs. Expressed in such a way, rather than referring to a computational procedure like SLD resolution, they are more easily extendable to other logic languages.

However, the definition proposed in [2] for correct answers was based on the model theoretic idea that a correct answer for an existentially quantified formula $\exists\vec{x}.\varphi$ is a substitution θ for the variables in $\vec{x}$ such that $\varphi\theta$ is true. When we tried to extend this simple idea from the well known case of Horn clauses to the full first order logic, we came with a general definition of correct answers which was rather involved and far less general than expected.

Since most of the work in the field of logic programming are heavily based on computed answers, which are the *computational* counterpart of correct answers, defining a solid foundation for the latter is essential if we want to adapt previous results to broader fragments of logic.

Here, we tackle the problem of correct answers from a proof theoretic point of view. We argue that the natural extension of the idea of "correct answer" to the first order logic is the recording of all the occurrences of quantifier introduction rules in a proof. In the case of Horn clauses, if we only consider the introductions of existential quantifiers, this turns out to be equivalent to the standard definition. We also introduce a corresponding generalization for "correct resultants". Then, we consider a common abstraction for the semantics of logic programs, namely *groundness*, and we examine its extension to the case of full first order logic. A prototypical abstract interpreter for this observable has been developed, and we show some of the results we have obtained. At last, possible future developments are discussed.

2 Preliminaries

We recall here the basic ideas which lie behind our framework. A more detailed treatment of these topics can be found in [2].

2.1 Basic Definitions

Logics can be presented in several different ways: we will stick here to a Gentzen-like proof-theoretic formulation. Given a set $\mathcal{S}$ of *sequents*, consider the untyped term signature Σ whose constant symbols are the elements of $\mathcal{S}$, provided with an overloaded symbol tree which has all the finite arities greater than 0. We call *proof skeleton* a term over Σ and we denote by $\mathsf{Sch}(\mathcal{S})$ the set T_Σ of all the proof skeletons. Then, we define two functions $\mathsf{hyp} : \mathsf{Sch}(\mathcal{S}) \to \mathcal{S}^\star$ and $\mathsf{root} : \mathsf{Sch}(\mathcal{S}) \to \mathcal{S}$ such that

- $\mathsf{hyp}(S) = S \quad ,$
- $\mathsf{hyp}(\mathsf{tree}(S, T_1, \ldots, T_n)) = \mathsf{hyp}(T_1) \cdots \mathsf{hyp}(T_n) \quad ,$

and

- $\mathsf{root}(S) = S \quad ,$
- $\mathsf{root}(\mathsf{tree}(S, T_1, \ldots, T_n)) = S \quad .$

Given a proof skeleton π, $\mathsf{hyp}(\pi)$ is the sequence of *hypotheses* of π while $\mathsf{root}(\pi)$ is the *root* of π. When we want to state that π is a proof skeleton with $\mathsf{hyp}(\pi) = S_1, \ldots, S_n$ and $\mathsf{root}(\pi) = S$, we write

$$\pi : S_1, \ldots, S_n \vdash S \ . \tag{2.1}$$

We also define the *height* of a proof skeleton π introducing the function $\mathsf{height} : \mathsf{Sch}(\mathcal{S}) \to \mathbb{N}$ such that

- $\mathsf{height}(S) = 0 \quad ,$
- $\mathsf{height}(\mathsf{tree}(S, T_1, \ldots, T_n)) = \max\{\mathsf{height}(T_1), \ldots, \mathsf{height}(T_n)\} + 1 \quad ,$

with the obvious assumption that $\max(\emptyset) = 0$.

Note that S, which we also denote by ϵ_S, and $\mathsf{tree}(S)$ are two different proof skeletons. Actually, it is $\mathsf{height}(S) = 0$ and $\mathsf{hyp}(S) = S$ but $\mathsf{height}(\mathsf{tree}(S)) = 1$ and $\mathsf{hyp}(\mathsf{tree}(S)) = \lambda$.

Now, we fix a set $\mathcal{R}$ of proof skeletons of height one. We call *inference rules* the elements of $\mathcal{R}$. A proof skeleton π, which is obtained by pasting together the empty proof skeletons and the inference rules, is called *proof.* A proof with no hypothesis is said to be *final.* A sequent S is *provable* if there is a final proof rooted at S. Finally, we call *sequent calculus* a pair $(\mathcal{S}, \mathcal{R})$.

2.2 Semantics

In the following, we assume fixed a sequent calculus $(\mathcal{S}, \mathcal{R})$. Given a sequent S, we denote by Sch_S the set of all the proof skeletons rooted at S. For each $\pi \in \mathsf{Sch}_S$ of the form

$$\pi : S_1, \ldots, S_n \vdash S \ , \tag{2.2}$$

we have a corresponding semantic operator $\pi : \mathsf{Sch}_{S_1} \times \cdots \times \mathsf{Sch}_{S_n} \to \mathsf{Sch}_S$ which works by pasting proof skeletons of the input sequents together with π, to obtain a new proof skeleton of the output sequent S. If Sch is the set of all the proof skeletons, $\pi : S_1, \ldots, S_n \vdash S \in \mathsf{Sch}$ and $X_i \subseteq \mathsf{Sch}$ for each i, we define a collecting variant of the semantic operator π, defined as

$$\pi(X_1, \ldots, X_n) = \{\pi(\pi_1, \ldots, \pi_n) \mid \forall i.\ \pi_i \in X_i \cap \mathsf{Sch}_{S_i}\} \ . \tag{2.3}$$

We will write $\pi(X)$ as a short form for $\pi(X, \dots, X)$ with n identical copies of X as input arguments.

An *intepretation* for $(\mathcal{S}, \mathcal{R})$ is a subset of Sch. We denote by $\mathcal{I}$ the set of all the interpretations, which is a complete lattice under subset ordering. A *model* is an interpretation I such that, for each inference rule $r \in \mathcal{R}$, it is

$$r(I) \subseteq I \ . \tag{2.4}$$

Models form a complete lattice under the same ordering of the interpretations. However, it is not a sublattice, since the join operator and the bottom element differ. In particular, the bottom element of the lattice of models is what we call *declarative semantics* of the sequent calculus and we denote it by $\mathcal{D}$.

$\mathcal{D}$ turns out to be the set of final proofs of $(\mathcal{S}, \mathcal{R})$. Hence, the declarative semantics precisely captures all the terminating computations. For a valid treatment of compositionality, we also need information about partial computations [5]. If ϵ is the set of all the empty proof schemas, we call *complete declarative semantics* of $(\mathcal{S}, \mathcal{R})$ and we denote it by $\mathcal{D}_c$, the least model greater then ϵ. It is possible to prove that $\mathcal{D}_c$ is actually the set of all the proofs of $(\mathcal{S}, \mathcal{R})$.

We have a bottom-up and a top-down construction of the least models using a couple of operators, similar in spirit to the immediate consequence operator T_P and the unfolding operator U_P for logic programs. The bottom-up T operator, mapping interpretations to interpretations, is defined as follows

$$T(I) = I \cup \bigcup_{r \in \mathcal{R}} r(I) \ , \tag{2.5}$$

while the top-down U operator is

$$U(I) = \bigcup_{\pi \in I} \pi(\mathcal{R} \cup \epsilon) \ . \tag{2.6}$$

The following properties hold:

$$T^{\omega}(\emptyset) = \mathcal{D} \ , \tag{2.7}$$

$$T^{\omega}(\epsilon) = \mathcal{D}_c = U^{\omega}(\epsilon) \ . \tag{2.8}$$

2.3 Abstractions

It is now possible to use the techniques of abstract interpretation [10] to develop a range of abstract semantics for sequent calculi. We call *observable* a triple $(\mathcal{A}, \alpha, \gamma)$ where $\mathcal{A}$ (the *abstract domain*) is an ordered set w.r.t. the relation $\sqsubseteq$ and $\alpha : \mathcal{I} \to \mathcal{A}$ (the *abstraction function*) is a monotonic function with γ as right adjoint. Since α and γ in $(\mathcal{A}, \alpha, \gamma)$ uniquely determine each other [11], we will often refer to an observable just by the abstraction function.

Given an observable and an operator on interpretations $\otimes$, we denote by $\otimes_\alpha$ the corresponding optimal abstract operator on $\mathcal{A}$. All the common results of the theory of abstract interpretation are inherited by our framework. In particular, the following properties hold:

$$T_\alpha^{\omega}(\alpha(\emptyset)) \sqsupseteq \alpha(\mathcal{D}) \ , \tag{2.9}$$

$$T_\alpha^\omega(\alpha(\epsilon)) \sqsupseteq \alpha(\mathcal{D}_c) \ , \tag{2.10}$$
$$U_\alpha^\omega(\alpha(\epsilon)) \sqsupseteq \alpha(\mathcal{D}_c) \ . \tag{2.11}$$

This means we can compute an approximation of the abstract semantics $\alpha(\mathcal{D})$ working entirely within the abstract domain $\mathcal{A}$.

Following the terminology introduced in [8] and [1], when T_α is precise (i.e. when $T_\alpha \circ \alpha = \alpha \circ T$) the observable α is said to be *denotational.* In this case, the "*greater than*" sign in the equations (2.9) and (2.10) can be replaced by an equality sign. When U_α is precise, the observable is *operational* and the equation (2.11) become an equality. When α is both operational and denotational, it is called *perfect.*

3 Correct Answers and Resultants

The concepts of *correct answer* and *computed answer* are the cornerstones of the theory and practice of logic programming. If we want to extend the results we have for Horn clauses to other logic languages, we need to find an analogous of these concepts in the new settings. Actually, since we are not discussing any computational mechanism, we only focus our attention to correct answers.

For the sake of clarity, we will assume to work in the domain of first order classical and intuitionistic logic. Therefore, we have a term signature Σ, a predicate signature Π and an infinite set V of variables. Terms and formulas are defined as usual, sequents are made of two sequences of formulas, separated by the symbol $\twoheadrightarrow$, and inference rules are those obtained as instances of the schemas in Figure 1. When we want to denote a particular inference rule, we index the name of the schema in figure with the formulas occurring in the instance. For example, $\wedge L_{\lambda,\varphi,\varphi',\psi,\Delta}$ is

$$\frac{\varphi, \varphi' \twoheadrightarrow \psi, \Delta}{\varphi \wedge \varphi' \twoheadrightarrow \psi, \Delta} \ . \tag{3.1}$$

Intuitionistic logic differs from classical one by allowing at most one formula in the right hand side of the sequents. The common abbreviation $\exists x_1, \ldots, x_n.\varphi$ stands for $\exists x_1. \cdots \exists x_n.\varphi$. We also write $\vec{\exists}\varphi$ and $\vec{\forall}\varphi$ for the existential and universal closure of φ. A *Horn sequent* is a sequent of the form $\Gamma \twoheadrightarrow \vec{\exists}\varphi$ where Γ is a sequence of universal closures of definite clauses, and φ is a definite goal.

3.1 Correct Answers as Proof-Theoretical Properties

Given a Horn sequent $S = \Gamma \twoheadrightarrow \exists\vec{x}.\varphi$, according to the standard definition, a correct answer for the goal $\exists\vec{x}.\varphi$ in the program Γ is a substitution θ for $\vec{x}$ such that $\Gamma \twoheadrightarrow \varphi\theta$ is provable. In the following, we refer to θ as a correct answer for the sequent S. According to this definition, the concept of correct answer seems strictly related to model theory. It is essentially an assignment for the variables in $\vec{x}$ such that φ in valid in every Herbrand model of Γ.

However, if π is an intuitionistic proof for S, a correct answer for $\exists\vec{x}.\varphi$ can be extracted from π by examining the instances in π of the $\exists R$ schema.

$$\frac{\Gamma_1, B, C, \Gamma_2 \twoheadrightarrow \Delta}{\Gamma_1, C, B, \Gamma_2 \twoheadrightarrow \Delta}\ \mathit{interchangeL} \qquad \frac{\Gamma \twoheadrightarrow \Delta_1, B, C, \Delta_2}{\Gamma \twoheadrightarrow \Delta_1, C, B, \Delta_2}\ \mathit{interchangeR}$$

$$\frac{\Gamma, B, B \twoheadrightarrow \Delta}{\Gamma, B \twoheadrightarrow \Delta}\ \mathit{contractionL} \qquad \frac{\Gamma \twoheadrightarrow B, B, \Delta}{\Gamma \twoheadrightarrow B, \Delta}\ \mathit{contractionR}$$

$$\frac{}{\Gamma, B \twoheadrightarrow B, \Delta}\ \mathit{id} \text{ where } B = \bot \text{ or } B \text{ is an atomic formula} \qquad \frac{\Gamma \twoheadrightarrow \bot}{\Gamma \twoheadrightarrow \Delta}\ \bot R$$

$$\frac{\Gamma \twoheadrightarrow B, \Delta}{\Gamma \twoheadrightarrow B \vee C, \Delta}\ \vee R_1 \qquad \frac{\Gamma \twoheadrightarrow B, \Delta}{\Gamma \twoheadrightarrow C \vee B, \Delta}\ \vee R_2 \qquad \frac{\Gamma, B \twoheadrightarrow D, \Delta \qquad \Gamma, C \twoheadrightarrow D, \Delta}{\Gamma, B \vee C \twoheadrightarrow D, \Delta}\ \vee L$$

$$\frac{\Gamma, B_1, B_2 \twoheadrightarrow C, \Delta}{\Gamma, B_1 \wedge B_2 \twoheadrightarrow C, \Delta}\ \wedge L \qquad \frac{\Gamma \twoheadrightarrow B, \Delta \quad \Gamma \twoheadrightarrow C, \Delta}{\Gamma \twoheadrightarrow B \wedge C, \Delta}\ \wedge R$$

$$\frac{\Gamma \twoheadrightarrow B, \Delta \quad \Gamma, C \twoheadrightarrow \Delta}{\Gamma, B \supset C \twoheadrightarrow \Delta}\ \supset L \qquad \frac{\Gamma, B \twoheadrightarrow C, \Delta}{\Gamma \twoheadrightarrow B \supset C, \Delta}\ \supset R$$

$$\frac{\Gamma, B[x/t] \twoheadrightarrow \Delta}{\Gamma, \forall x.B \twoheadrightarrow \Delta}\ \forall L \qquad \frac{\Gamma \twoheadrightarrow B[x/a], \Delta}{\Gamma \twoheadrightarrow \forall x.B, \Delta}\ \forall R \text{ where } a \text{ is a fresh variable}$$

$$\frac{\Gamma, B[x/a] \twoheadrightarrow \Delta}{\Gamma, \exists x.B \twoheadrightarrow \Delta}\ \exists L \text{ where } a \text{ is a fresh variable} \qquad \frac{\Gamma \twoheadrightarrow B[x/t], \Delta}{\Gamma \twoheadrightarrow \exists x.B, \Delta}\ \exists R$$

Figure 1: Inference rules for classical logic

Example 3.1 If S is the sequent $p(0), \forall x.p(x) \twoheadrightarrow \exists y.p(y)$, then y/t is a correct answer for each term t. If π is the proof

$$\frac{\dfrac{}{\forall x.p(x), p(0) \twoheadrightarrow p(0)}\ \mathit{id}}{\forall x.p(x), p(0) \twoheadrightarrow \exists y.p(y)}\ \exists R \tag{3.2}$$

the $\exists R$ rule gives origin to the correct answer $y/0$. ■

When we use hereditary Harrop formulas, we can keep the same definition of correct answers we have for Horn clauses. However, the amount of information we obtain in this way is rather limited. For example, the sequent $\forall x.p(x,x) \twoheadrightarrow \forall y.\exists z.p(y,z)$, only has a trivial empty correct answer, since the right hand side of the sequent is not an existentially quantified formula. On the contrary, let us give a look to a proof of the same sequent:

$$\frac{\dfrac{\dfrac{\dfrac{}{p(a,a) \twoheadrightarrow p(a,a)}\ \mathit{id}}{p(a,a) \twoheadrightarrow \exists z.p(a,z)}\ \exists R}{\forall x.p(x,x) \twoheadrightarrow \exists z.p(a,z)}\ \forall R}{\forall x.p(x,x) \twoheadrightarrow \forall y.\exists z.p(y,z)}\ \forall L \tag{3.3}$$

If we keep track of the occurrences of both the $\exists R$ and $\forall R$ inference rules, we obtain a substitution $\{y/a, z/a\}$. This makes explicit that for each y we have a

z such that $p(y,z)$ is true, and that y and z do coincide. Moreover, if we apply the substitution $\{y/a, z/a\}$ to the right hand side, discarding all the quantifiers, we obtain the sequent $\forall x.p(x,x) \twoheadrightarrow p(a,a)$ which is trivially provable.

If we further extend the language to handle the full first order logic, we have to treat with sequents like $\exists x.p(s(x)) \twoheadrightarrow \exists y.p(y)$. Here, again, the standard "model-theoretic" definition of correct answers gives us no interesting information, since there are no correct answers according to that definition. Actually, the sequent $\exists x.p(s(x)) \twoheadrightarrow p(t)$ is not provable for any term t. Let us consider the following proof:

$$\dfrac{\dfrac{\dfrac{}{p(s(a)) \twoheadrightarrow p(s(a))}\,id}{p(s(a)) \twoheadrightarrow \exists y.p(y)}\,\exists R}{\exists x.p(s(x)) \twoheadrightarrow \exists y.p(y)}\,\exists L \tag{3.4}$$

If we keep track of all the instances of a quantifier introduction rule, we obtain a substitution $\{x/a, y/s(a)\}$. Here, the role of a is that of a witness. The existential quantifier on the left hand side *produces* a new object a such that $p(s(a))$ holds. The binding $\{y/s(a)\}$ makes clear that the object y such that $p(y)$ holds is $s(a)$, where a is the same produced by the other existential quantifier. Again, if we apply the substitution discarding the quantifiers, we obtain the sequent $p(s(a)) \twoheadrightarrow p(s(a))$ which is provable.

3.2 Formalization

We now try to make precise the above informal discussion. Given a first order language (Σ, Π, V), a *candidate answer* is a function $\theta : V \to \mathcal{P}_f(T_\Sigma(V))$ such that $\{v \mid \theta(v) \neq \emptyset\}$ is finite. We denote with Ans the set of candidate answers.

For each proof skeleton π, we have a corresponding candidate answer θ_π or $\mathsf{answer}(\pi)$, defined as follows:

$$\theta_\pi(x) = \begin{cases} \emptyset & \text{if } \pi = id_{\Gamma,B,\Delta} \\ \{a\} & \text{if } \pi = \exists L_{\Gamma,B,\Delta,x,a}(\pi') \text{ or } \pi = \forall R_{\Gamma,B,\Delta,x,a}(\pi') \\ \{t\} & \text{if } \pi = \exists R_{\Gamma,B,\Delta,x,t}(\pi') \text{ or } \pi = \forall L_{\Gamma,B,\Delta,x,t}(\pi') \\ \bigcup_{j=1\ldots n} \theta_{\pi_j}(x) & \text{if } \pi = r(\pi_1, \ldots, \pi_n) \end{cases} \tag{3.5}$$

If π is a proof, then θ_π is called the *partial correct answer* for π. If π is a final proof of the sequent S, then θ is a *correct answer* for the sequent S. The set of correct answers for the sequent S will be denoted by $\mathsf{CAns}(S)$, which is defined as

$$\mathsf{CAns}(S) = \{\theta_\pi \mid \pi \in \mathfrak{D} \cap \mathsf{Sch}_S\} \ . \tag{3.6}$$

We write $\mathsf{CAns}_c(S)$ or $\mathsf{CAns}_i(S)$ when we want to make clear if we are working in the realm of classical or intuitionistic logic.

Example 3.2 Let us consider the sequent $S = \forall x.(p(x) \supset p(s(x))), p(0) \twoheadrightarrow \exists y.p(y)$. In classical logic, θ is a correct answer for S if and only if

- $\theta(x) \in \mathcal{P}_f(\mathsf{Term})$,

- $\theta(y) \in \mathcal{P}_f(\mathsf{Term})$ and there exists $s^i(0) \in \theta(y)$ such that $s^j(0) \in \theta(x)$ for every $j \in \{0, \ldots, i-1\}$.

In intuitionistic logic, the form of the correct answers is simpler. In particular, θ is a correct answer for S if and only if

- $\theta(x) \in \mathcal{P}_f(\mathsf{Term})$,
- $\theta(y) = \{s^i(0)\}$ for some $i \in \mathbb{N}$ such that $s^j(0) \in \theta(x)$ for every $j \in \{0, \ldots, i-1\}$.

The difference is due to the fact we cannot apply the contraction rule on the consequent. ■

Note that a correct answer for the sequent S is meaningful only if there are no two different bindings for the same variable. However, explicitly requiring this condition would have made the definitions more complex. Therefore, we have preferred this presentation. In the following, we will call *pure* every sequent which satisfies the above condition on bound variables.

Our definition of correct answers essentially collects all the occurrences of introduction rules for quantifiers in a proof. A problem is that most of the answers we obtain are trivial. For example, if x is a variable which is bound from a negative universal quantifier and θ is a correct answer, then $\theta[x/L]$ is a correct answer, too, for each $L = \theta(x) \cup T$ where T is a set of terms renamed apart from θ.

As a result, we are particularly interested to *minimal* correct answers, according to the obvious point-wise ordering. Proofs corresponding to minimal answers are a sort of "non-redundant" proofs, where quantifiers are introduced only when they are really needed. In formulas, we denote by $\mathsf{mAns}_c(S)$ ($\mathsf{mAns}_i(S)$) the set of minimal correct answer for the sequent S w.r.t. classical (intuitionistic) logic.

Example 3.3 In the previous example, both classic and intuitionistic logic have the same set of minimal correct answers, i.e. those θ such that

- $\theta(y) = \{s^i(0)\}$ for some $i \in \mathbb{N}$,
- $\theta(x) = \{s^j(0) \mid j \in \{0, \ldots, i-1\}\}$.

Note that we have a lot of information from these. We know that $p(s^i(0))$ is true for every $i \in \mathbb{N}$. Moreover, we know that, in order to prove $p(s^i(0))$, we need to apply a $\forall L$ introduction rule for the first binding with different terms, namely all the $s^j(0)$ for j from 0 to $i-1$. ■

In general, if $\mathsf{mAns}_c(S) \neq \mathsf{mAns}_i(S)$, it means that there is a proof of S which is "intrinsically" classical. We do not make precise this statement, since it requires further investigations. However, from an intuitive point of view, consider the following proof π of the sequent $p(a) \vee p(b) \twoheadrightarrow \exists x.p(x)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p(a) \twoheadrightarrow p(a), p(b)}\,id \qquad \dfrac{}{p(b) \twoheadrightarrow p(a), p(b)}\,id}{p(a) \vee p(b) \twoheadrightarrow p(a), p(b)}\,\vee L}{p(a) \vee p(b) \twoheadrightarrow \exists x.p(x), \exists x.p(x)}\,\exists R \text{ 2 times}}{p(a) \vee p(b) \twoheadrightarrow \exists x.p(x)}\,\text{contraction}R \tag{3.7}$$

If we move the $\exists R$ rules upward, before the $\vee L$ rule, we can easily obtain an intuitionistic proof π' such that $\theta_{\pi'} \leq \theta_\pi = \{x/\{a,b\}\}$. However, consider the following proof π

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p(a), p(b) \twoheadrightarrow p(b), \bot}\ id}{p(a), p(b) \twoheadrightarrow \exists x.p(x), \bot}\ \exists R}{p(a), \exists y.p(y) \twoheadrightarrow \exists x.p(x), \bot}\ \exists L}{p(a) \twoheadrightarrow \exists y.p(y) \supset \bot, \exists x.p(x)}\ \supset R}{p(a) \twoheadrightarrow \exists x.p(x) \vee (\exists y.p(y) \supset \bot),\ \exists x.p(x) \vee (\exists y.p(y) \supset \bot)}\ \vee R \text{ (2 times)}}{p(a) \twoheadrightarrow \exists x.p(x) \vee (\exists y.p(y) \supset \bot)}\ \text{contraction}R \tag{3.8}$$

Although the root sequent is intuitionistically provable, we are not able to write an intuitionistic proof π' such that $\theta_{\pi'} \leq \theta_\pi = \{x/b, y/b\}$. This is because the use of the contraction rule in π is essential to the effort of moving $\exists y.p(y)$ on the left side while keeping $\exists x.p(x)$ on the right side.

If we compare the “standard” correct answers for Horn clauses with our minimal correct answers, we still have a more general definition. However, if we restrict our answers to the existential quantifiers, we obtain the same results.

Theorem 3.4 *If $S = \Gamma \twoheadrightarrow \exists x_1. \cdots \exists x_n.\varphi$ is a pure Horn sequent, then η is a “standard” correct answer for S iff there is a minimal correct answer θ such that $\theta(x_i) = x_i\eta$ for each $i \in \{1, \ldots, n\}$.*

Note that we have not specified if the minimal correct answer should be considered w.r.t. intuitionistic or classical logic. Actually, if S is a pure Horn sequent, it is $\mathsf{mAns}_i(S) = \mathsf{mAns}_c(S)$.

3.3 Resultants

Another typical abstraction of SLD-derivations is those of resultants [13]. A resultant for a goal `G` in a program `P` is a pair made of a partial computed answer for `G` and a new goal `G'` which still needs to be refuted. We present an observable for proof skeletons which is inspired by this “standard” idea of resultant, although the relation here is more shallow than for correct answers.

Until now we have considered sequents as sequences of formulas. However, classical and intuitionistic logics are often presented by defining a sequent as a set of formulas. We use the term *set sequent* to refer to this alternative definition and we denote by $\mathsf{Set}\mathcal{S}$ the collection of all the set sequents. If $S \in \mathcal{S}$, we write $\bar{S}$ for the corresponding element in $\mathsf{Set}\mathcal{S}$.

We call *resultant* a pair $(\theta, \mathcal{S})$ where $\theta \in \mathsf{Ans}$ and $\mathcal{S}$ is a finite multi-set of set sequents. We denote by Res the set of all the resultants. For each proof skeleton $\pi : S_1, \ldots, S_n \vdash S$, we define a corresponding $\mathsf{res}(\pi)$ as

$$\mathsf{res}(\pi) = (\theta_\pi, \llbracket \bar{S_1}, \ldots, \bar{S_n} \rrbracket) \ , \tag{3.9}$$

where $\Lbag \_ \Rbag$ denote a multi-set. If π is a proof for the sequent S, then $\mathsf{res}(\pi)$ is a *correct resultant* for S. The set of correct resultants for S will be denoted by $\mathsf{CRes}(S)$, which is defined as

$$\mathsf{CRes}(S) = \{\mathsf{res}(\pi) \mid \pi \in \mathcal{D}_c \cap \mathsf{Sch}_S\} \ . \tag{3.10}$$

We write $\mathsf{CRes}_c(S)$ or $\mathsf{CRes}_i(S)$ when we want to specify if we are working in the realm of classical or intuitionistic logic. We define an order relation between two resultants, according to the following equation

$$(\theta, \mathcal{S}) \leq (\theta', \mathcal{S}') \text{ iff } \theta \leq \theta' \text{ and } \mathcal{S} \subseteq \mathcal{S}' \ . \tag{3.11}$$

Again, we talk of *minimal* correct resultants for the elements of $\mathsf{CRes}(S)$ which are minimal w.r.t. $\leq$. We denote the corresponding sets as $\mathsf{mRes}_c(S)$ and $\mathsf{mRes}_i(S)$.

Example 3.5 Let us consider the sequent $S = p(0) \twoheadrightarrow \exists x.p(x)$. The set $\mathsf{mRes}_c(S)$ contains all the pairs $(\theta, \mathcal{S})$ such that

- $\theta = \{x/0\}$ and $\mathcal{S} = \emptyset$, or
- $\theta = \{x/t\}$ for $t \neq 0$ and $\mathcal{S} = \Lbag p(0) \twoheadrightarrow p(t) \Rbag$, or
- $\theta = \{x/t\}$ for $t \neq 0$ and $\mathcal{S} = \Lbag p(0) \twoheadrightarrow \bot \Rbag$.

The same happens for $\mathsf{mRes}_i(S)$. ■

It is trivial to prove that the following correspondences hold between correct answers and correct resultants:

$$\mathsf{CAns}(S) = \{\theta \mid (\theta, \emptyset) \in \mathsf{CRes}(S)\} \ , \tag{3.12}$$
$$\mathsf{mAns}(S) = \{\theta \mid (\theta, \emptyset) \in \mathsf{mRes}(S)\} \ . \tag{3.13}$$

4 Observables

Now that we have defined what a correct answer is, we would like to find a bottom-up and a top-down construction for CAns and mAns. Following the abstract framework in [2], we define the observable of candidate answers as a tuple $\langle [\mathcal{S} \to \mathcal{P}(\mathsf{Ans})], \alpha_c, \gamma_c \rangle$ where $[\mathcal{S} \to \mathcal{P}(\mathsf{Ans})]$ is the set of functions from sequents to sets of candidate answers and

$$\alpha_c(I)(S) = \{\theta_\pi \mid \pi : \cdot \vdash S \in I\} \ . \tag{4.1}$$

It is trivial to show that $\alpha_c(\mathcal{D})(S)$ is exactly the set $\mathsf{CAns}(S)$ of all the correct answers for the sequent S. The optimal abstract operator corresponding to T is T_{α_c}. Assuming A in the image of α_c, it is

$$T_{\alpha_c}(A)(S) = A(S) \cup \bigcup \{r_{\alpha_c}(A) \mid r \in \mathcal{R}, \mathsf{root}(r) = S\} \ , \tag{4.2}$$

where, for each $r : S_1 \ldots S_n \vdash S \in \mathcal{R}$,

$$r_{\alpha_c}(A) = \{r_{\alpha_c}(\theta_1, \ldots, \theta_n) \mid \forall i \in \{1, \ldots, n\}, \theta_i \in A(S_i)\} \ , \tag{4.3}$$

and

$$r_{\alpha_c}(\theta_1, \dots, \theta_n) = \begin{cases} \theta_1 \cup [x/t] & \text{if } r \text{ is an introduction rule for a quantifier} \\ & \text{which replaces the variable } x \text{ with } t, \\ \theta_1 \cup \dots \cup \theta_n & \text{otherwise.} \end{cases} \tag{4.4}$$

Here we write $\theta_1 \cup \theta_2$ for the candidate answer θ such that $\theta(x) = \theta_1(x) \cup \theta_2(x)$ for each $x \in V$ and $[x/t]$ for the candidate answer θ such that $\theta(x) = \{t\}$ and $\theta(y) = \emptyset$ for $y \neq x$.

Theorem 4.1 *The observable α_c of candidate answers is denotational.*

We also have an observable for *resultants* which gives origin to complete bottom-up and top-down semantics. It is defined as the tuple $\langle[\mathsf{Set}\mathcal{S} \to \mathcal{P}(\mathsf{Res})], \alpha_r, \gamma_r\rangle$ with the abstraction function

$$\alpha_r(I)(\bar{S}) = \{\mathsf{res}(\pi) \mid \pi \in I, \mathsf{root}(\pi) = S', \bar{S} = \bar{S}'\} \ . \tag{4.5}$$

The definition of the optimal bottom-up fixpoint operator is straightforward. With respect to the top-down fixpoint operator, assuming A in the image of α_r, we have

$$U_{\alpha_c}(A)(\bar{S}) = \bigcup_{\delta \in A(\bar{S})} , \delta(\mathcal{R} \cup \epsilon) \tag{4.6}$$

where, if $\delta = (\theta, \llbracket \bar{S}_1, \dots, \bar{S}_n \rrbracket)$,

$$\delta(X) = \{\delta(\llbracket \pi_1, \dots, \pi_n \rrbracket) \mid \forall i \in \{1 \dots n\}, \pi_i \in X \cap \mathsf{Sch}_{S'_i} \text{ and } \bar{S}'_i = \bar{S}_i\} \ , \tag{4.7}$$

and

$$\delta(\llbracket \pi_1, \dots, \pi_n \rrbracket) = \Big(\theta \cup [x_1/t_1] \cup \dots \cup [x_m/t_m], \llbracket \overline{\mathsf{hyp}(\pi_1)}, \cdots, \overline{\mathsf{hyp}(\pi_n)} \rrbracket\Big) \ , \tag{4.8}$$

where, for each pair (x_i, t_i), there is an introduction rule for quantifiers among $\{\pi_1, \dots, \pi_n\}$ which replaces the variable x_i with t_i.

Theorem 4.2 *The observable α_r of correct resultants is perfect.*

Since α_r is a perfect observable, we could build a top-down interpreter which computes correct resultants. However, the efficient implementation of such an interpreter is a very difficult task which is the realm of automatic deduction. Here, we are more interested in computing abstractions of correct resultants, which can be used for static analysis of logic languages.

4.1 Groundness

If θ is a candidate answer for the sequent S, we say that θ is *grounding* for the variable x when $\theta(x)$ only contains variables which occur free in S. Let us define by GAns the set of functions $V \to \mathcal{P}(\{\mathrm{g}, \mathrm{ng}\})$, which we call *groundness answers.* Given a candidate answer θ for the sequent S, we define a corresponding groundness answer $\beta = \mathsf{ground}_S(\theta)$ such that

- g $\in \beta(x)$ iff there exists $t \in \theta(x)$ such that $\mathrm{vars}(\theta) \subseteq \mathrm{FV}(S)$,
- ng $\in \beta(x)$ iff there exists $t \in \theta(x)$ such that $\mathrm{vars}(\theta) \not\subseteq \mathrm{FV}(S)$,

where $\mathrm{FV}(S)$ is the set of free variables in S. Then, we can define a Galois connection $\langle \alpha_g, [\mathcal{S} \twoheadrightarrow \mathcal{P}(\mathsf{Ans})], [\mathcal{S} \twoheadrightarrow \mathcal{P}(\mathsf{GAns})], \gamma_g \rangle$ where

$$\alpha_g(A)(S) = \{\mathsf{ground}_S(\theta) \mid \theta \in A(S)\} \ . \tag{4.9}$$

Then, by composing α_g with α_c, we obtain an observable $\langle \mathcal{P}(\mathsf{GAns}), \alpha_g \circ \alpha_c, \gamma_c \circ \gamma_g \rangle$ for groundness answers. If $\beta \in \alpha_g(\mathsf{CAns}(S))$, then β is a *correct* groundness answer. Moreover, if $\beta \in \alpha_g(\mathsf{mAns}(S))$, then β is *minimal* according to the obvious point-wise ordering.

Example 4.3 Let us give some examples of sequents and their corresponding correct minimal groundness answers for intuitionistic logic.

sequent	**groundness answers**
$\forall y.p(y) \twoheadrightarrow \exists x.p(x)$	$\{x/g, y/g\}$ $\{x/ng, y/ng\}$
$\forall y.(p(a,y) \wedge p(y,b)) \twoheadrightarrow \exists x.p(x,x)$	$\{x/g, y/g\}$
$p(a) \vee r(b) \twoheadrightarrow \exists x.(p(x) \vee r(x))$	$\{x/g\}$
$\bot \twoheadrightarrow \exists x.p(x)$	$\{\}$
$\forall y.p(y,y) \twoheadrightarrow \forall x_1.\exists x_2.p(x_1,x_2)$	$\{y/g, x_1/g, x_2/g\}, \{y/ng, x_1/ng, x_2/ng\}$
$\forall x_1.\ \exists x_2.\ p(x_1,x_2) \twoheadrightarrow \exists y.p(y,y)$	—
$\exists y.p(y) \twoheadrightarrow \exists x.p(x)$	$\{x/ng, y/ng\}$
$p(t(a)) \twoheadrightarrow \exists x.p(r(x))$	—
$p(a) \vee \exists x.r(x) \vdash \exists y.(p(y) \vee r(y))$	$\{x/ng, y/\{g, ng\}\}$

■

Note that if θ is a correct answer for S and x is a bound variable which only appears in a negative existential quantifier, then θ is not grounding for x. The same happens for positive universal bindings.

We may ask ourselves which is the correspondence between our observable and standard domains for analysis of groundness such as $\mathcal{POS}$ [4]. It is possible to prove the following

Theorem 4.4 *Let* $\mathtt{P}$ *be a definite program and* $\mathtt{G}$ *a definite goal. We work in the realm of intuitionistic logic. Assume* $S = \Gamma \twoheadrightarrow \exists x_1, \ldots, x_n.\ G$ *is the corresponding pure Horn sequent. Consider* $x_1, \ldots, x_n$ *as propositional symbols and define the formula*

$$\Theta = \bigvee_{\beta \in \mathsf{GAns}(S)} \{\wedge_i x_i^{\beta(x_i)}\} \ , \tag{4.10}$$

where $x_i^{\{\mathrm{g}\}} = x_i$ *and* $x_i^{\{\mathrm{ng}\}} = \neg x_i$. *Then* Θ *is a positive formula.*

Moreover, if X *is the set of correct answers of* $\mathtt{G}$ *in* $\mathtt{P}$, *let* $\aleph = \alpha_{\mathcal{POS}}(X)$. *Then* $\aleph$ *and* Θ *are equivalent formulas.*

We can also build an abstraction for groundness analysis starting from resultants. Given a formula φ, we denote by $\alpha_v(\varphi)$ an *abstract formula* obtained from φ by replacing each term with the set of free variables occurring in them. Let us call by $\mathsf{G}\mathcal{S}$ the set of *abstract set sequents* obtained from abstract formulas. A *groundness resultant* is a pair (β, ν) such that $\beta \in \mathsf{GAns}$ and ν is a multi-set of elements of $\mathsf{G}\mathcal{S}$. We write as GRes the set of groundness resultants. Then, we can define a Galois connection $\langle \alpha_{rg}, [\mathsf{Set}\mathcal{S} \to \mathcal{P}(\mathsf{Res})], [\mathsf{G}\mathcal{S} \to \mathcal{P}(\mathsf{GRes})], \gamma_{rg} \rangle$, where

$$\begin{aligned} \alpha_{rg}(A)(\mathsf{S}) = \{(\mathsf{ground}_S(\theta), \{\alpha_v(\bar{S_1}), \ldots, \alpha_v(\bar{S_n})\}\} \mid \\ (\theta, \{\bar{S_1}, \ldots, \bar{S_n}\}) \in A(\bar{S}), \mathsf{S} = \alpha_v(\bar{S})\} \ . \end{aligned} \tag{4.11}$$

We can compose α_{rg} with α_r to obtain a new observable which is well suited for a top-down analysis of *correct groundness resultants*. Following this idea, we have developed a prototypical abstract interpreter for intuitionistic logic. It is written in `PROLOG` and can be found at the URL `http://www.di.unipi.it/~amato/papers/`. Actually, it computes *minimal* correct groundness answers, since this often reduces the huge amount of abstract sequents which are computed otherwise.

Example 4.5 By applying our analyzer to the sequents in the Example 4.3 we obtain precisely the same set of correct groundness answers, with the following exceptions:

sequent	**groundness answers**
$p(t(a)) \twoheadrightarrow \exists x.p(r(x))$	$\{x/ng\}$
$\forall x_1.\ \exists x_2.\ p(x_1, x_2) \twoheadrightarrow \exists y.p(y, y)$	$\{y/ng, x_1/ng, x_2/ng\}$

■

The previous example shows two different situations in which we loose precision. The first one is due to the fact that we abstract a term with the set of its free variables, discarding information about the functors. The second situations arises from the fact that the abstract domain is not enough powerful to keep track of the side condition for the $\exists L$ and the $\forall R$ introduction rules. To overcome this problem, we would need to improve the representation of abstract terms, by introducing a sort of *labelling* similar to what [17] does for hereditary Harrop formulas.

5 Conclusions and Future Works

In this paper we have presented a new definition for correct answers and correct resultants which can be applied to the full first order logic (both classical and intuitionistic). Moreover, we have shown that a well known abstraction of logic program semantics, namely groundness, can be easily reintroduced inside our framework. This definitions are so general that they can be reused with only slight changes for every logic system with standard quantifier rules, such as linear logic or modal logic. We think that, w.r.t. [2], our new definitions of correct answers and groundness answers give us more intuitive and accurate results and a much cleaner theory.

From the point of view of the implementation of abstract domains, several things can be improved in the framework. For example, while a top-down analyzer can often be implemented straightforwardly, like our interpreter for groundness, the same definitely does not hold for bottom-up analyzers. Since for a bottom-up analysis we have to build the entire abstract semantics of a sequent calculus, we need a way to isolate a finite number of "representative sequents" from which the semantics of all the others can easily be inferred: it is essentially a problem of compositionality.

We are actually studying this problem and we think that extending the notion of a sequent calculus with the introduction of some *rules for the decomposition of sequents* will add to the theoretical framework the power needed to easily derive compositional T operators.

We also need a way to reduce nondeterminism in abstract interpreters. This is a problem which has been tackled thoroughly in the field of automatic deduction. A standard solution is to use unification to reduce nondeterminism in the introductions of quantifiers [6, 19]. We would like to treat unification in our framework, and we want to do this without any major modification. We are working in the direction of defining an abstraction of proof skeletons using extra-logical variables such that the corresponding optimal abstract operators automatically computes the semantics trough unification.

References

[1] G. Amato and G. Levi. Properties of the lattice of observables in logic programming. In M. Falaschi and M. Navarro, editors, *Proceedings of the APPIA-GULP-PRODE'97 Joint Conference on Declarative Programming*, pages 175–187, 1997.

[2] G. Amato and G. Levi. Abstract Interpretation Based Semantics of Sequent Calculi. In J. Palsberg, editor, *Static Analysis Symposium 2000*, volume 1824 of *Lecture Notes in Computer Science*, pages 38–57. Springer-Verlag, 2000.

[3] J. M. Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

[4] T. Armstrong, K. Marriott, P. Schachte, and H. Søndergaard. Boolean functions for dependency analysis: Algebraic properties and efficient representation. In B. Le Charlier, editor, *Proc. Static Analysis Symposium, SAS'94*, volume 864 of *Lecture Notes in Computer Science*, pages 266–280. Springer-Verlag, 1994.

[5] A. Bossi, M. Gabbrielli, G. Levi, and M. C. Meo. A Compositional Semantics for Logic Programs. *Theoretical Computer Science*, 122(1–2):3–47, 1994.

[6] K.A. Bowem. Programming with full first-order logic. *Machine Intelligence*, pages 421–440, 1982.

[7] A. Brogi, P. Mancarella, D. Pedreschi, and F. Turini. Modular logic programming. *ACM Transactions on Programming Languages and Systems*, 16(4):1361–1398, July 1994.

[8] M. Comini, G. Levi, and M. C. Meo. A theory of observables for logic programs. *Information and Computation*, 1999. To appear.

[9] M. Comini, G. Levi, and G. Vitiello. Modular abstract diagnosis. In *International Workshop on Tools and Environments for (Constraint) Logic Programming, ILPS'97 Postconference Workshop*, 1997.

[10] P. Cousot and R. Cousot. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In *Proc. Fourth ACM Symp. Principles of Programming Languages*, pages 238–252, 1977.

[11] P. Cousot and R. Cousot. Abstract Interpretation and Applications to Logic Programs. *Journal of Logic Programming*, 13(2 & 3):103–179, 1992.

[12] S. K. Debray. Formal bases for dataflow analysis of logic programs. In G. Levi, editor, *Advances in logic programming theory*, pages 115–182. Clarendon Press, Oxford, 1994.

[13] M. Gabbrielli, G. Levi, and M. C. Meo. Resultants semantics for PROLOG. *Journal of Logic and Computation*, 6(4):491–521, 1996.

[14] S.C. Kleene. Permutability of inferences in Gentzen's calculi LK and LJ. *Memoris of the AMS*, 10, 1952.

[15] J. W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, 1987. Second edition.

[16] D. Miller, F. Pfenning, G. Nadathur, and A. Scedrov. Uniform proofs as a foundation for Logic Programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.

[17] G. Nadathur. A Proof Procedure for the Logic of Hereditary Harrop Formulas. *Journal of Automated Reasoning*, 11:115–145, 1993.

[18] G. Nadathur and D. Miller. An Overview of λProlog. In Kenneth A. Bowen and Robert A. Kowalski, editors, *Fifth International Logic Programmiong Conference*, pages 810–827. MIT Press, 1988.

[19] N. Shankar. Proof search in the intuitionistic sequent calculus. In M.E. Stickel, editor, *10th Int. Conf. on Automated Deduction*, volume 449 of *Lecture Notes in Artifical Intelligence*, pages 522–536. Springer-Verlag, 1990.

A Proofs

Lemma A.1 *If the pure sequent $S = \Gamma \twoheadrightarrow \exists x.\varphi$ has a correct answer θ such that $\theta(x) = \emptyset$, then $\Gamma \twoheadrightarrow \bot$ is provable.*

Proof. If π is a final proof of S and $\theta_\pi(x) = \theta(x) = \emptyset$, then there are no occurrences of $\exists R$ inference rules, with $\exists x.\varphi$ as the principal formula, in π. However, looking at the form of inference rules in Figure 1, it is evident that we need a way to discard the existential quantifier from the right hand side of the sequent and the only way, other than an $\exists R$ rule, is a $\bot R$ rule.

Then, for each path in π from the root to the leaves, either $\exists x.\varphi$ is never the principal formula of an introduction rule, or a $\bot R$ inference rule is applied. We can obtain a new proof π' of $\Gamma \twoheadrightarrow \bot$ just replacing every occurrence of $\exists x.\varphi$ with $\bot$, and deleting the $\bot R$ rules which are now useless. ■

Proof of theorem 3.4. Assume η is a standard correct answer for S. It means that $S' = \Gamma \twoheadrightarrow \varphi\eta$ is provable. Let π' be a proof of S'. If we apply a series of $\exists R$ rules to π', we obtain a proof π for S. It is trivial that $\theta_\pi(x_i) = \{x_i\eta\}$ for each $i \in \{1, \ldots, n\}$. Now, consider the set of all the correct answers θ' for S such that $\theta' \leq \theta_\pi$. If $\theta'(x_i) \neq \theta_\pi(x_i)$, then $\theta'(x_i) = \emptyset$, since $\theta(x_i)$ is a singleton. By the previous lemma, however, this would mean that Γ is inconsistent, and this is not possible for Horn clauses. Then, if we take θ to be a minimal θ', we prove half of the theorem.

Now, assume θ is a minimal correct answer for S. We want to prove that, if we define $\eta(x_i) = \theta(x_i)$ for $i \in \{1, \ldots, n\}$, then $\Gamma \twoheadrightarrow \varphi\eta$ is provable. If π is a proof such that $\theta = \theta_\pi$, we can think of permuting the inference rules to obtain a new proof π' with $\theta_{\pi'} = \theta$ and all the $\exists R$ rules applied just after the root. Since in Horn clauses we do not have positive universal quantifiers or negative existential quantifiers, in π there are no occurrences of $\forall R$ or $\exists L$ rules. As a result, in π there are no eigenvariables. Therefore, rules in classical logic can be permuted freely to obtain π'. If we work in intuitionistic logics, not all the permutations are allowed, but Kleene in [14] shows that the only rules $\exists R$ does not permute with are $\vee L$ and $\exists L$. Neither of this can never be applied if Γ is made of Horn clauses, hence π' can be found in intuitionistic logic, too. If we drop the $\exists R$ rules from π', we remain with a proof of $\Gamma \twoheadrightarrow \varphi\eta$ and the theorem is proved. ■

Proof of Theorem 4.1. We need to prove that

$$T_{\alpha_c}(\alpha_c(I)) \subseteq \alpha_c(T(I)) \ , \tag{A.1}$$

since the opposite disequality is trivial.

Assume $\theta \in \mathsf{Ans}$ and $\theta \in T_{\alpha_c}(\alpha_c(I))(S)$. We have two cases: $\theta \in \alpha_c(I)(S)$ or $\theta \in r_{\alpha_c}(\alpha_c(I))$ for some $r : S_1, \ldots, S_n \vdash S \in \mathcal{R}$. If $\theta \in \alpha_c(I)(S)$, then $\theta \in \alpha_c(T(I))(S)$ follows trivially. Otherwise, it is $\theta = r_{\alpha_c}(\theta_1, \ldots, \theta_n)$, with $\theta_i \in \alpha_c(I)(S_i)$ for each $i \in \{1, \ldots, n\}$.

If r is an introduction rule for a quantifier, which replaces the variable x with the term t, then $\theta = \theta_1 \cup [x/t]$. Since $\theta_1 \in \alpha_c(I)(S_1)$, there exists a final proof skeleton in I with $\theta_1 = \theta_\pi$. By applying the rule r to π, we obtain a new final proof skeleton $\pi' : \cdot \vdash S$ such that $\theta = \theta_{\pi'}$. Since $\pi' \in T(I)$, it is $\theta \in \alpha_c(T(I))(S)$.

If r is not and introduction rule for a quantifier, then $\theta = \theta_1 \cup \ldots \cup \theta_n$. For each θ_i, there exists a proof $\pi_i : \cdots \vdash S_i$ in I. By applying the rule r to π, we can reason as in the previous case, and we prove the theorem. ■

Proof of Theorem 4.2. We need to prove that α_r is both operational and denotational. We only prove it is operational, since the other proof proceeds as for Theorem 4.1. Actually, we only need to check the disequality

$$U_{\alpha_r}(\alpha(I)) \subseteq \alpha_r(U(I)) \ , \tag{A.2}$$

since the opposite one is trivial.

If $\delta = (\theta, \llbracket \bar{S_1}, \ldots, \bar{S_n} \rrbracket) \in U_{\alpha_r}(\alpha_r(I))(S)$, there exists $\delta' = (\theta', \llbracket \bar{S'_1}, \ldots, \bar{S'_m} \rrbracket)$ in $\alpha_r(I)(S)$ such that $\delta = \delta'(\llbracket r_1, \ldots, r_m \rrbracket)$, where $r_i \in \mathcal{R} \cup \epsilon$ for each $i \in \{1, \ldots, m\}$. By the definition of α_r, there is a proof $\pi' : Z'_1, \ldots, Z'_l \vdash Z$ in I with $Z, Z'_i \in \mathcal{S}$, $\bar{Z} = \bar{S}$, $\llbracket \bar{Z'_1}, \ldots, \bar{Z'_l} \rrbracket = \llbracket \bar{S'_1}, \ldots, \bar{S'_m} \rrbracket$ and $\theta_{\pi'} = \theta'$. We can apply π' to appropriate $r'_1, \ldots, r'_n$ such that $\llbracket \bar{r_1}, \ldots, \bar{r_n} \rrbracket = \llbracket \bar{r'_1}, \ldots, \bar{r'_n} \rrbracket$ to obtain a proof $\pi : Z_1, \ldots, Z_r \vdash Z \in U(I)$ with $\llbracket \bar{Z_1}, \ldots, \bar{Z_r} \rrbracket = \llbracket \bar{S_1}, \ldots, \bar{S_m} \rrbracket$ and $\theta_\pi = \theta$. This proves the theorem. ■

Proof of Theorem 4.4. First of all, since Horn clauses are always consistent, it follows from Lemma A.1 that, if β is a correct groundness answer for S, then $\theta(x_i) \neq \emptyset$. Moreover, since we work in intuitionistic logic, each $\exists R$ inference rule can be applied only once for each variable, hence $\beta(x_i)$ is a singleton for each x_i. Therefore, $x_i^\beta(x_i)$ is well defined.

Now, consider an ordering $\sqsubseteq$ on $\{g, ng\}$ such that $g \sqsubseteq ng$ with the corresponding lifting to groundness answers. If $S = \Gamma \twoheadrightarrow \exists x_1, \ldots, x_n.\ G$ is a pure Horn sequent and β is a correct groundness answer, we call *existential* groundness answer the restriction of β to $\{x_1, \ldots, x_n\}$. We denote by $\mathsf{EAns}(S)$ the set of all the existential groundness answer for the sequent S. If $\beta \in \mathsf{EAns}(S)$, then $\beta' \in \mathsf{EAns}(S)$ for each $\beta' \sqsubseteq \beta$. Therefore, consider the formula Θ. If S is a provable, there is an existential groundness answer β and, by the previous property, the answer $\beta' \sqsubseteq \beta$ such that $\beta'(x_i) = \{g\}$ for each i is an element of EAns. Hence Θ is positive.

We still need to prove that Θ and $\aleph$ are equivalent. Given an assignment ν of truth values to $x_1, \ldots, x_n$, Θ is true iff there is an existential answer β such that $\beta(x_i) = \{g\}$ if $\nu(x_i) = \mathsf{true}$, $\beta(x_i) = \{ng\}$ otherwise. In turn, this means that there exists a correct answer θ such that $\mathrm{vars}(\theta(x_i)) \nsubseteq \mathrm{FV}(S)$ for each i with $\nu(x_i) = \mathsf{false}$. Since $\mathrm{FV}(S) = \emptyset$, it means that there is a correct answer θ such that $\theta(x_i)$ is not ground for each $x_i \in \nu^{-1}(\mathsf{false})$.

By the definition of $\alpha_{\mathcal{POS}}$, we have that $\aleph$ is true under the assignment ν iff for each correct answers θ, $\theta(x_i)$ is ground if $\nu(x_i) = \mathsf{true}$. But this is equivalent to state that there exists a correct answer θ such that $\nu(x_i) = \mathsf{false}$ implies $\theta(x_i)$ not ground. Actually, this is the same statement which holds for Θ, then we have proved the required equivalence. ■