INFORMATIQUE THÉORIQUE ET APPLICATIONS

J. VAUZEILLES
A. STRAUSS

Intuitionistic three-valued logic and logic programming

Informatique théorique et applications, tome 25, nº 6 (1991), p. 557-587.

<http://www.numdam.org/item?id=ITA_1991__25_6_557_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 25, n° 6, 1991, p. 557 à 587)

INTUITIONISTIC THREE-VALUED LOGIC AND LOGIC PROGRAMMING (*)

by J. VAUZEILLES (1) and A. STRAUSS (2)

Communicated by J. E. PIN

Abstract. – *In this paper, we study the semantics of logic programs with the help of trivalued logic, introduced by Girard in* 1973.

Trivalued sequent calculus enables to extend easily the results of classical SLD-resolution to trivalued logic. Moreover, if one allows negation in the head and in the body of Horn clauses, one obtains a natural semantics for such programs regarding these clauses as axioms of a theory written in the intuitionistic fragment of that logic.

Finally, we define in the same calculus an intuitionistic trivalued version of Clark's completion, which gives us a declarative semantics for programs with negation in the body of the clauses, the evaluation method being SLDNF-resolution.

Résumé. – *Dans cet article, nous étudions la sémantique des programmes logiques à l'aide de la logique trivaluée, introduite par Girard en* 1973.

Le calcul des séquents trivalué permet d'étendre très simplement les résultats de la SLD-résolution classique à la logique trivaluée. De plus, si on considère des clauses de Horn en autorisant l'utilisation de la négation dans la tête et le corps des clauses, une sémantique naturelle de ces programmes est obtenue en considérant les clauses du programme comme les axiomes d'une théorie dans le fragment intuitionniste de cette logique.

Enfin, nous définissons, toujours dans le calcul, une version trivaluée intuitionniste du complété de Clark, ce qui nous permet d'obtenir une sémantique déclarative pour les programmes avec négation dans le corps des clauses, la méthode d'évaluation étant la SLDNF-résolution.

I. INTRODUCTION

In this paper we investigate the links between three-valued logic and logic programming; instead of the more usual model-theoretical approach ([4, 10]) we adopt a proof-theoretical viewpoint, like Gallier's [5] for classical logic. For this, we use the three-valued sequent calculus introduced by Girard [7],

(*) Received November 1989, revised December 1990.

(1) L.I.P.N., Université Paris-Nord et C.N.R.S., avenue Jean-Baptiste-Clément, 93430 Villetaneuse, France.

(2) Université Paris-VII, C.N.R.S. U.A. 753, place Jussieu, 75007 Paris.

0988-3754/91/06 557 31/$5.10/

that is a three-valued version of Gentzen's sequent calculus [6], and particularly its intuitionistic subsystem.

In section II, we recall some results on **SLD**-resolution and in appendix A, we give the proofs of these results using Gentzen's sequent calculus to show how we can extend them to three-valued logic.

In section III.1 and appendix B, we define the notions of three-valued models and Girard's three-valued logic.

In sections III.2 and III.3 we show that we can easily extend the resolution method and **SLD**-resolution to three-valued logic; the results of these sections are very close to those of Schmitt [12].

In section III.4 we show the completeness and the soundness of the extension of **SLD**-resolution to clauses containing negative literals (also in the head of the clauses) with respect to intuitionistic three valued logic; many expert systems use such resolution and so, it is useful to work out an exact semantics for these systems. Delahaye announced close results, without proofs, and using Fitting's approach in [3].

In section IV, we study negation as failure: we define for each program **P**, **Comp∗ (P)**, which is a three-valued version in Girard's logic of the Clark's completion. Though Fitting [4] and Kunen [10] use three-valued logie to define semantics for negation as failure, our approach is different.

We show the soundness of **SLDNF**-resolution with respect to **Comp∗ (P)** in intuitionistic three-valued logic as suggested by Shepherdson [15] (for programs with variables) and the completeness in the propositional case.

For another axiomatization of negation as failure, using Girard's linear logic, *see* [1].

II. SLD-RESOLUTION

In this section, we recall some results on **SLD**-resolution. We suppose that the reader is familiar with these notions. If not, he can consult [5] or [11]. The results that we want to extend to three-valued logic are proved in appendix A: we adopt a proof-theoretical viewpoint and so, we use Gentzen's sequent calculus and we show what results are intuitionistically valid.

If a formula A is classically (resp. intuitionistically) provable in a classical (resp. intuitionistic) theory T, we write $\mathbf{T} \vdash A$ (resp. $\mathbf{T} \vdash_{\mathrm{I}} A$).

II.1. Definition: (i) A **clause** is a formal expression $\Gamma \vdash \Delta$, Γ and Δ being sequences (possibly empty) of atomic formulae; the meaning of a clause,

$A_1, \ldots, A_n \vdash B_1, \ldots, B_p$ is, as usual, $A_1 \& \ldots \& A_n \rightarrow B_1 \vee \ldots \vee B_p$ (*see* A.1.3).

(ii) A **Horn clause** is a clause $\Gamma \vdash \Delta$ such that Δ contains at most one atom.

(iii) A **definite clause** is a Horn clause $\Gamma \vdash \Delta$ where Δ contains (exactly) one formula. A definite clause where Γ is empty is said to be a **positive** clause.

(iv) A **negative clause** is a Horn clause $\Gamma \vdash \Delta$ such that Δ is empty.

(v) An **anti-Horn clause** is a formula A where A is a conjunction $B_1 \& \ldots \& B_n$, each B_i being a literal, and at most one B_i being negative. The **associated-clause** of A is the Horn clause $B_1, \ldots, B_{i-1}, B_{i+1}, \ldots, B_n \vdash C_i$ if $B_i = \neg C_i$ is the negative literal, otherwise it is $B_1, \ldots, B_n \vdash$ (if there is no negative literal).

II.2. DEFINITION: A **logic program** is a pair (**P**, **Q**) where **P** is a set of Horn clauses, and **Q** a formula of the form $A_1 \vee \ldots \vee A_q$, each A_i being an anti-Horn clause. We say that **Q** is the query and we consider for **SLD**-refutations the set of clauses obtained by adding to clauses of **P** the associated-clause of A_i for each A_i. We denote the existential closure of **Q** by $\exists \mathbf{Q}$.

Remark that if $\mathbf{Q} = \neg H_1 \vee \ldots \vee \neg H_q$, where each $H_i (= \neg A_i)$ is a Horn-clause and if $\mathbf{P} \vdash \mathbf{Q}$, then the inconsistency of $\mathbf{P} \cup \{H_1, \ldots, H_q\}$ may be checked by *SLD*-resolution, because all clauses in this set are indeed Horn clauses.

We are interested in this form of programs because, in our generalization to clauses containing negation in their head (part IV), naturally negative *N*-clauses appear.

II.3. THEOREM (completeness): *let* (**P**, **Q**) *be a logic program; then,*

(1) *either* **P** *is* **inconsistent**, *and there exists at least one* **SLD**-*refutation of* **P** *with goal a negative clause of* **P**; *no other negative clause is used in the refutation;*

(2) *or* **P** *is* **consistent** *and then,*

(i) *if* **Q** *is a disjunction* $A_1 \vee \ldots \vee A_q$, *each* A_i *being a conjunction of atoms, and if* $\mathbf{P} \vdash \exists \mathbf{Q}$, *then* $\mathbf{P} \vdash_I \mathbf{Q}\theta$ (*for some substitution* θ) *and* **Q** *succeeds with answer including* θ *under all computation rules* (*using* **SLD**-*resolution*);

(ii) *if* **Q** *is a disjunction of anti-Horn clauses, and if* $\mathbf{P} \vdash \exists \mathbf{Q}$, *then there exists a sequence of substitutions* $\theta_1, \ldots, \theta_n$ *such that* $\mathbf{P} \vdash \mathbf{Q}\theta_1 \vee \ldots \vee \mathbf{Q}\theta_n$ *and* **Q** *succeeds* (*under* **SLD**-*resolution*) *with indefinite answers including* $\theta_1, \ldots, \theta_n$.

Proof: See **appendix A**: theorem A.4.1; (ii) is first proved in [5]. Remark that if **P** is a set of definite clauses, it is always consistent.

II.4. THEOREM (soundness): (i) *Let* **(P, Q)** *be a logic program with* $\mathbf{Q} = A_1 \vee \ldots \vee A_q$ *(each* A_i *being an anti-Horn clause). Let* π *be a* **SLD-resolution** *and* $(C_1\theta_1), \ldots, (C_n\theta_n)$ *the substitutions of associated-clauses of formulae* $A_1, \ldots, A_q$ *used in* π, *then* $\mathbf{P} \vdash \mathbf{Q}\theta_1 \vee \ldots \vee \mathbf{Q}\theta_n$.

(ii) *Moreover, if* **P** *is a set of definite clauses, if each* A_i *is a conjunction of atoms, and if* θ *is the result substitution, then* $\mathbf{P} \vdash_I \ni \mathbf{Q}$ *and* $\mathbf{P} \vdash_I \mathbf{Q}\theta$.

Proof: See [5] for (i); (ii) only one negative clause is used in the refutation, and this clause is the associated-clause of A_i for some i $(1 \leqq i \leqq q)$; then, by (i) $A_1\theta \vee \ldots \vee A_q\theta$ is a logical consequence of **P** and by lemma A.3.4 of **appendix A**, it is intuitionistically derivable from **P**; then $\mathbf{Q}\theta$ and $\ni \mathbf{Q}$ are intuitionistically derivable from **P**.

III. THREE-VALUED LOGIC

III.1. Kleene's three-valued structures and Girard's three-valued logic

In this section we recall some definitions and results on three-valued models and three-valued logic. The reader can find more details, references and proofs, in [7] and in [9].

III.1.1. DEFINITION: Let **L** be a fixed first-order language. A **three-valued structure M** for **L** consists of the following data:

(i) a non-empty set $|\mathbf{M}|$, the domain of **M**;

(ii) for each n-ary function letter f of **L**, a function $\mathbf{M}(f): |\mathbf{M}|^n \to |\mathbf{M}|$;

(iii) for each n-ary predicate p of **L**, a function $\mathbf{M}(p): |\mathbf{M}|^n \to \{\mathbf{t}, \mathbf{u}, \mathbf{f}\}$ (*i.e.* **true, undetermined, false).**

– A three-valued structure is **binary** if it does not take the value **u** (hence, it is a binary structure in the familiar sense).

We shall denote by **L**[**M**] the language obtained by adding to **L** new constants $\underline{c}$, for each $c \in |\mathbf{M}|$.

III.1.2. DEFINITION: (i) If t is a closed term of **L**[**M**] one defines its value $\mathbf{M}(t)$ by: $\mathbf{M}(\underline{c}) = c$; $\mathbf{M}(ft_1 \ldots t_n) = \mathbf{M}(f)\,(\mathbf{M}(t_1) \ldots \mathbf{M}(t_n))$.

(ii) If A is a closed formula of **L**[**M**], one defines its value $\mathbf{M}(A)$ by:

(1) $\mathbf{M}(pt_1 \ldots t_n) = \mathbf{M}(p)\,(\mathbf{M}(t_1) \ldots \mathbf{M}(t_n))$;

(2) the values of formulae beginning with a propositional connective are defined according to the following truth tables:

A	B	$A \& B$	$A \vee B$	$A \to B$	$\neg A$
t	t	t	t	t	f
u	t	u	t	t	u
f	t	f	t	t	t
t	u	u	t	u	
u	u	u	u	u	
f	u	f	u	t	
t	f	f	t	f	
u	f	f	u	u	
f	f	f	f	t	

(3) – if $\mathbf{M}(A(\underline{c}))$ is true (resp. false) for some $c \in |\mathbf{M}|$, then $\mathbf{M}(\ni x, A(x)) = t$ [resp. $\mathbf{M}(\forall x, A(x)) = f$];

– if $\mathbf{M}(A(\underline{c}))$ is true (resp. false) for all $c \in |\mathbf{M}|$, then $\mathbf{M}(\forall x, A(x)) = t$ [resp. $\mathbf{M}(\ni x, A(x)) = f$];

– in the other cases $\mathbf{M}(\ni x, A(x))$ and $\mathbf{M}(\forall x, A(x))$ take the value u.

III.1.3. Definition: (i) The **language 3 L** is defined as follows:

terms: the terms of **L**.

formulae: the normal expressions $?A$ and $!A$, where A is a formula of L. We shall represent an arbitrary formula of **3L** by ξA, ηA, where ξ, η vary through the set $\{?, !\}$. We use the symbol $^\circ$ as follows: $\xi^\circ = !$ if $\xi = ?$, and $\xi^\circ = ?$ if $\xi = !$.

(ii) A sequent in **3L** is a formal expression $\Gamma \vdash \Delta$, where Γ and Δ are finite sequences (possibly empty) of formulae in **3L**.

III.1.4. Definition: (i) The closed formula ξA of **3L**[**M**] is valid in the three-valued structure **M** iff

– $\xi = !$ and $\mathbf{M}(A) = t$ (we say that A is necessary in **M**);

– $\xi = ?$ and $\mathbf{M}(A) \neq f$ (we say that A is possible in **M**).

We shall denote this fact by $M \models \xi A$ (and $M \not\models \xi A$ otherwise).

(ii) The closed sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ of **3L**[**M**] is valid in the three-valued structure **M** iff:

– if $n \neq 0$ and $m \neq 0$: if $M \models A_1$ and ... and $M \models A_n$ then $M \models B_1$ or ... or $M \models B_m$;

– if $n \neq 0$ and $m = 0$ then $M \not\models A_1$ or ... or $M \not\models A_n$;

– if $n = 0$ and $m \neq 0$ then $M \models B_1$ or ... or $M \models B_m$.

– if $n = 0$ and $m = 0$ the sequent $\vdash$ means absurdity.

III.1.5. DEFINITION: Let **S** be a set of sequents of **3L**. A **three-valued model M** of **S** is a three-valued structure where any closed instance of a sequent of **S** is valid in **M**. **S** is **3-consistent** if it has at least one three-valued model.

If not, we say that **S** is **3-inconsistent**.

If all closed instances of a formula A of **3L** are valid in any three-valued model of a set **S** of sequents of **3L**, we say that A is a (three-valued) logical consequence of **S**.

Girard has defined the sequent calculus **3LK** (resp. **3LI**) in the spirit of Gentzen's calculus **LK** (resp. **LI**); these systems enjoy completeness and soundness with respect to classical (resp. intuitionistic) three-valued models and can be extended to second order logic (*see* [7] and [9]). We present these systems in appendix B. For the definition of intuitionistic three-valued models (topological or Kripke three-valued models) *see* [7].

If a formula A or a sequent $\Gamma \vdash \Delta$ is provable in the theory **S** in $3\,LK$ (resp. $3\,LI$), we write $\mathbf{S} \vdash_{3\,K} \mathbf{A}$ or $\mathbf{S} \vdash_{3\,K} \Gamma \vdash \Delta$ (resp. $\mathbf{S} \vdash_{3\,I} \mathbf{A}$ or $\mathbf{S} \vdash_{3\,I} \Gamma \vdash \Delta$).

III.2. 3-resolution

III.2.1. DEFINITION: A **3-clause** is a sequent of **3L**, containing only formulae of the form $!A$, $?A$, A being an atom (we say that the formula ξA is a **3-atom**).

III.2.2. DEFINITION: – We define the language **L2** as follows: variables, constants, functions symbols are those of **L**. For each predicate letter of **L**, we introduce two predicate letters $p!$ and $p?$ with the arity of p.

– For each formula A of **3L**, we define a formula $(A)_2$ of **L2** (and we say that $(A)_2$ is the **2-translated** of A), as follows:

$$(\xi p(t_1, \ldots, t_n))_2 = (p\,\xi)(t_1, \ldots, t_n)$$

$$(\xi \neg A)_2 = \neg (\xi^\circ A)_2, \qquad (\xi A \,\&\, B)_2 = (\xi A)_2 \,\&\, (\xi B)_2$$

$$(\xi A \vee B)_2 = (\xi A)_2 \vee (\xi B)_2$$

$$(\xi A \rightarrow B)_2 = (\xi^\circ A)_2 \rightarrow (\xi B)_2$$

$$(\xi \ni x, A(x))_2 = \ni x (\xi A(x))_2$$

$$(\xi \forall x, A(x))_2 = \forall x (\xi A(x))_2$$

– For each sequent $\Gamma \vdash \Delta$ of **3L**, we define its 2-translated $(\Gamma \vdash \Delta)_2$: it is the sequent obtained by replacing any formula occurring in the sequent by its **2**-translated.

– If **M** is a three-valued structure, we define a binary structure **M2** for **L2** as follows: the interpretation of the constants and of the function symbols

does not change. As for the predicate symbols, we define

$$\mathbf{M2} \models (p\xi)(t_1 \ldots t_n) \quad \text{iff } \mathbf{M} \models \xi p(t_1 \ldots t_n)$$

III.2.3. LEMMA: *Let* **R** *be the set of all* 3-*clauses*

$$!p(x_1, \ldots, x_n) \vdash ?p(x_1, \ldots, x_n) \quad (\textit{for each predicate } p).$$

Let **S** *be a set of sequents of* **3L**, **S2** *the set of the* 2-*translated of the sequents of* **S**.

(i) *if* **M** *is a three-valued model of* **S**, **M2** is a binary model of **S2** $\cup$ **R2**.

(ii) if **N** *is a binary model of* **S2** $\cup$ **R2**, then there exists a three-valued model **M** *of* **S** *with* **N** = **M2**.

Proof: Immediate.

III.2.4. LEMMA: *Let* A *be a formula of* **3L** *and* **S** *be a set of sequents of* **3L**, *then there exists a set* **T** *of* 3-*clauses which satisfies the following proposition:*

A *is a logical consequence of* **S** *iff* **T** *is* 3-*inconsistent.*

Proof: We replace the set **S** and the formula A by their 2-translated **S2** and $(A)_2$; A is a three-valued consequence of **S** iff $(A)_2$ is a logical consequence of **S2** $\cup$ **R2**; we construct (as usual) a set **T**′ of 2-clauses so that: **T**′ is inconsistent iff $(A)_2$ is a logical consequence of **S2** $\cup$ **R2**. Let **T** be the set of 3-clauses so that **T2** = **T**′ − **R2**; using lemma III.2.3, we obtain the result.

III.2.5. DEFINITION: We define **3-resolution** just like ordinary resolution (which we designate by 2-resolution), but the clauses involved are 3-clauses instead of ordinary clauses; remark that a 3-atom $!A$ (resp. $?A$) can be unified with a 3-atom $!B$ (resp. $?B$) iff A and B can be unified. A 3-atom $!A$ cannot be unified with a 3-atom $?B$.

III.2.6. THEOREM (soundness and completeness): *Let* **S** *be a set of* 3-*clauses and* **S**° *the set obtained by adding to* **S** *the following* 3-*clauses* $!p \vdash ?p$ (*for each predicate* p *occurring in* **S**).

Then **S** *is* 3-*inconsistent iff the sequent* $\vdash$ *is provable with* 3-*resolution from clauses of* **S**°.

(*In fact it is enough to add to* **S** *only clauses* $!p \vdash ?p$ *for each predicate* p *such that* $!p$ *appears in the right part of a clause, and* $?p$ *appears in the left part of a clause.*)

Proof: It is an immediate consequence of soundness and completeness for resolution in the binary case and of lemma III.2.3: by lemma III.2.3, **S** is

3-inconsistent iff $\mathbf{S}^\circ\mathbf{2}$ is (classically) inconsistent; but, $\mathbf{S}^\circ\mathbf{2}$ is inconsistent iff $\vdash$ is provable by 2-resolution using the clauses of $\mathbf{S}^\circ\mathbf{2}$; then a proof of the sequent $\vdash$ using 2-resolution from $\mathbf{S}^\circ\mathbf{2}$ can easily be transformed in a proof of the sequent $\vdash$ using 3-resolution from $\mathbf{S}^\circ$ and vice-versa [replacing each predicate $(p\,\xi)$ by $\xi\,p$].

III.3. Horn 3-clauses

2-resolution is a very expensive process, and therefore, often, we only consider the subclass of Horn clauses. Similarly, for three-valued resolution, we will restrict our study to the class of Horn 3-clauses (defined below).

III.3.1. Definition: (i) A **Horn 3-clause** is a 3-clause such that the right part contains at most one 3-atom.

As examples,

$$!A(x, y), \qquad ?B(x, y) \vdash ?C(x, z) \tag{1}$$

$$!A(x, y), \qquad ?A(x, y) \vdash \tag{2}$$

(ii) A **definite 3-clause** is a Horn 3-clause where the right part contains (exactly) one formula (example 1).

(iii) A **negative 3-clause** is a Horn 3-clause where the right part is empty (example 2).

(iv) An **anti-Horn clause** is a formula ξA where A is a conjunction $B_1 \,\&\, \ldots \,\&\, B_n$, each B_i being a literal, and at most one B_i being negative. The **associated-clause** to ξA is the Horn clause $\xi B_1, \ldots, \xi B_{i-1}, \xi B_{i+1}, \ldots, \xi B_n \vdash \xi^\circ C_i$ if $B_i = \neg C_i$ is the negative literal, otherwise it is $\xi B_1, \ldots, \xi B_n \vdash$ (if there is no negative literal).

III.3.2. Definition: We define **3 SLD-resolution** (resp. **3 SLD-refutation**) just like **SLD**-resolution (resp. **SLD**-refutation) replacing clauses by 3-clauses.

III.3.3. Definition: A **3-logic program** is a pair **(P, Q)** where **P** is a set of 3-Horn clauses, and **Q** is a disjunction $\xi A_1 \vee \ldots \vee A_n$, each ξA_i being an anti-Horn 3-clause. We say that **Q** is the query and we consider for **3 SLD**-refutations the set of clauses obtained by adding to clauses of **P** the associated-clause of ξA_i (for each i), and clauses $!p \vdash ?p$ for each atom p.

(In fact it is enough to add to **S** only clauses $!p \vdash ?p$ for each predicate p such that $!p$ appears in the right part of a clause, and $?p$ appears in the left part of a clause.)

III.3.4. THEOREM (soundness and completeness): *Theorems* II.3 *and* II.4 *carry over to* 3*-logic programs provided we apply* **3SLD**-*resolution to the set above defined.*

Proof: For soundness: as in III.2.2, we replace each predicate ξp by $(p\,\xi)$ and use the result of II.3. For completeness: if we replace each formula by its 2-translated, we can use the results of appendix A, and then we easily draw the conclusions of the theorem; remark that, since we work in extensions of classical (or intuitionistic) sequent calculus to three-valued logic, all results remain true.

III.3.5. *Comparison with related work*

In [12], Schmitt investigates a three-valued logic; he considers two negations symbols ($\sim$ and $\neg$) and four kinds of literals: A, $\neg A$, $\sim A$ and $\sim\neg A$ which mean respectively "A is true", "A is false", "A is not true", "A is not false". Hence they respectively correspond, in our langage, to $!A$, $!\neg A$, $?\neg A$, $?A$.

He considers A and $\sim\neg A$ as positive literals (*i.e.* our positive literals), and $\sim A$ and $\neg A$ as negative literals (*i.e.* our negative literals); he defines the implication $A \supset B$ as an abbreviation of $\sim A \vee B$; a clause is a disjunction of literals and a Horn clause is a clause containing at most one positive literal.

He shows that the usual soundness and completeness results for *SLD*-resolution (a query being an existential conjunction of positive literals) carry over provided we apply *SLD*-resolution to the program augmented by all clauses $A \supset \sim\neg A$ for all atoms A.

A structure **M** is a model of a Schmitt Horn clause $A_1 \& \ldots \& A_n \supset B$ iff **M** is a model of the 3-clause $A_1{}^*, \ldots, A_n{}^* \vdash B^*$ obtained by replacing each positive literal A by the corresponding positive literal A^* in our language. Remark that, in this translation, the translated of $A \supset \sim\neg A$ is $!A \vdash ?A$.

But our result differs from the Schmitt's one in the form of the queries.

III.4. Horn N-clauses. Semantics for programs with negation

In this section, we study classical logic programs with negation. We allow the negation to appear also in the head of the clauses. We show that the natural semantics for these programs is three-valued semantics.

III.4.1. DEFINITION: We define the notions of **N-clause, Horn N-clause, definite N-clause, negative N-clause**, just as we have defined the similar notions

for clauses: for this, we replace in definition II.1 the term "atom" by "literal".

As examples,

$$A(x, y), \qquad \neg B(x, y) \vdash \neg C(x, z)$$

is a definite N-clause;

$$A(x, y), \qquad \neg A(x, y) \vdash$$

is a negative N-clause.

III.4.2. DEFINITION: — We define the language **LN** as follows: the variables, constant, function symbols are those of **L**. For each predicate letter of **L**, we introduce the predicate letter p^* with the arity of p.

— If $\Gamma \vdash \Delta$ is a N-clause, and if we replace in this clause any negative literal $\neg p(t_1, \ldots, t_n)$ by $p^*(t_1, \ldots, t_n)$, we obtain a clause (in the familiar sense) of **LN**; we say that the clause obtained is the **N-translated** of $\Gamma \vdash \Delta$.

If in $\Gamma \vdash \Delta$, we replace any literal A by $!A$, we obtain a sequent of **3L**, and we say that this sequent is the **3-translated** of $\Gamma \vdash \Delta$.

— If **M** is a three-valued structure, we define a binary structure **MN** for **LN**, as follows:

$$\mathbf{MN} \models p(t_1, \ldots, t_n) \quad \text{iff } \mathbf{M} \models !p(t_1, \ldots, t_n)$$
$$\mathbf{MN} \models p^*(t_1, \ldots, t_n) \quad \text{iff } \mathbf{M} \models !\neg p(t_1, \ldots, t_n)$$

the interpretation of the constant and of the functions symbols being unchanged.

III.4.3. DEFINITION: Let **M** be a three-valued structure; we say that **M** is a three-valued model for a set **S** of N-clauses iff any 3-translated of a N-clause of **S** is valid in **M**. If **S** has a three-valued model, we say that **S** is 3-consistent; otherwise we say that **S** is 3-inconsistent.

III.4.4. LEMMA: *Let* **R** *be the set of all N-clauses of* **L** $p(x_1, \ldots, x_n)$, $\neg p(x_1, \ldots, x_n) \vdash$ (*for each predicate p of* **L**).

Let **S** *be a set of N-clauses,* **SN** (*resp. RN*) *the set of the N-translated of clauses of* **S** (*resp.* **R**), **S*** *the set of the 3-translated of clauses of* **S**.

(i) *If* **M** *is a three-valued model of* **S***, **MN** *is a binary model of* **SN** $\cup$ **RN**.

(ii) *If* **T** *is a binary model of* **SN** $\cup$ **RN**, *then there exists a three-valued model* **M** *of* **S*** *such that* **T** = **MN**.

Proof: Immediate.

III.4.5. DEFINITION: A **N-logic program** is a pair (**P**, **Q**) where **P** is a set of Horn N-clauses, and **Q** a formula of the form $A_1 \vee \ldots \vee A_n$, each A_i being a conjunction of literals: $A_i = B_{i\,1} \& \ldots \& B_{iq_i}$. Then, to construct **NSLD**-refutations, we add to **P** the N-Horn clauses $B_{i\,1}, \ldots, B_{iq_i} \vdash$ (for each i) and the clauses $p, \neg p \vdash$ (for each predicate p).

III.4.6. DEFINITION: We define **NSLD**-resolution just like resolution for Horn N-clauses; so, we can extend ordinary unification to literals (positive and negative).

III.4.7. THEOREM (completeness): *Let* (**P**, **Q**) *be a N-logic program; then,*

(1) *either* **P** *is* 3-*inconsistent, and there exists at least one* **NSLD**-*refutation of* **P** *with goal a negative N-clause of* **P** *or* **R** (*with the notations of lemma* III.4.4); *no other negative N-clause is used in the refutation;*

(2) *either* **P** *is* 3-*consistent and then, if* $P \vdash_{3K} \ni \mathbf{Q}$, *then* $\mathbf{P} \vdash_{3I} \ni \mathbf{Q}$ *and* $\mathbf{P} \vdash_{3I} \mathbf{Q}\theta$ (*for some substitution* θ) *and* **Q** *succeeds with answer including* θ. *Moreover no clause of the form* $p, \neg p \vdash$ *is used in the proof.*

Proof: (1) if **P** is a 3-inconsistent, then **PN** $\cup$ **RN** is (classically) inconsistent (lemma III.4.4); there exists a **SLD**-refutation; then we easily transform this **SLD**-refutation into a **NSLD**-refutation, replacing any p^* by $\neg p$;

(2) if **P** is 3-consistent then **PN** $\cup$ **RN** is (classically) consistent; then we can apply the results of section II to obtain a **SLD**-refutation that we easily transform into a **NSLD**-refutation. The fact that $\ni \mathbf{Q}$ and $\mathbf{Q}\theta$ are intuitionistically derivable from **P**, comes from the fact that the proofs of appendix A can be carried over Girard's three-valued logic (*see* appendix B).

III.4.8. THEOREM (soundness): *Let* (**P**, **Q**) *be a N-logic program with* $\mathbf{Q} = A_1 \vee \ldots \vee A_n$ (*each* A_i *being a conjunction of literals*). *Let* π *be a* **NSLD**-*refutation and* θ *the result substitution, then either* **P** *is* 3-*inconsistent and only a clause* $p, \neg p \vdash$ *is used in* π, *or* **P** *is* 3-*consistent and then, no clause* $p, \neg p \vdash$ *is used in the* **NSLD**-*refutation and* $\mathbf{P} \vdash_{3I} \ni \mathbf{Q}$ *and* $\mathbf{P} \vdash_{3I} \mathbf{Q}\theta$.

Proof: Replace any clause of the **NSLD**-refutation by its N-translated; we obtain a **SLD**-refutation. By results of section II, only one negative clause is used; if the negative clause used is $p, p^* \vdash$ for a predicate p) then **PN** $\cup$ **RN** is 2-inconsistent (with notations of definition III.3.4) and hence, **P** is 3-inconsistent (lemma III.4.4); if the negative clause is the N-translated of a clause $B_1, \ldots, B_q \vdash$ we easily obtain the result. To prove the intuitionistic derivability, we proceed just as in theorem II.4.

III.4.9. *Remark*

If we consider a 3-consistent set of Horn N-clauses **P** and if we add a positive N-clause $\vdash A$, A being a literal, a *NSLD* refutation of the above set of N-clauses corresponds to the query $?\ni B$ with $B = \neg A$ if A is a positive literal, and $A = \neg B$ otherwise; many substituted-clauses of $\vdash A$ can occur in the **NSLD**-refutation and then we obtain an indefinite answer (that is, substitutions $\theta_1, \ldots, \theta_q$ where $?B\theta_1 \vee \ldots \vee B\theta_q$ is a logical consequence of **P**).

III.4.10. *Comparison with related work*

In [3], Delahaye studies the semantics of programs with negation possibly in the body and in the head of clauses. He considers Kleene's three-valued connectives for $\neg$, &, $\vee$ and defines, as Kunen in [10], a new connective $\supset$, for implication: $A \supset B$ is false if A is true and B is false or undetermined, and $A \supset B$ is true otherwise. Using Fitting's methods [4], he announces correction and completeness results. Remark that **M** is a three-valued model of a Delahaye's clause $A_1 \& \ldots \& A_n \supset B$ ($A_1, \ldots, A_n$, B being literals) iff **M** is a three-valued model of our 3-translated $!A_1, \ldots, !A_n \vdash !B$ of the N-clause $A_1, \ldots, A_n \vdash B$.

IV. NEGATION AS FAILURE

In this section we define the completion **Comp* (P)** of a general program **P** with respect to intuitionistic 3-valued logic, in the spirit of Clark's completion **Comp (P)**; we show the soundness of **SLDNF**-resolution with respect to **Comp* (P)** in intuitionistic 3-valued logic.

If **P** is propositional, then we also obtain the completeness of **SLDNF**-resolution with respect to **Comp* (P)**.

We define as usual the notion of **SLDNF**-resolution: *see*, for example [11] or [15]; we use their terminology, particularly concerning computation rules.

We suppose that we have in **L** a predicate "$=$", whose intended interpretation is the identity relation.

IV.1. Definition: Let **P** be a normal program, that is a set of clauses

$$L_1, \ldots, L_q \vdash p(t_1, \ldots, t_n)$$

The **3-valued translation** of this clause is defined to be:

$$!L_1, \ldots, !L_q \vdash !p(t_1, \ldots, t_n)$$

and the **normal form** of this clause is defined to be:

$$\ni y_1 \ldots \ni y_k (x_1 = t_1 \,\&\, \ldots \,\&\, x_n = t_n \,\&\, L_1 \,\&\, \ldots \,\&\, L_q) \vdash p(x_1, \ldots, x_n)$$

if $y_1, \ldots, y_k$ are the variables of the original clause.

If the n-place predicate p occurs m times ($m > 0$) in the head of a clause in **P**, and if the normal form of these clauses are: $E_i \vdash p(x_1, \ldots, x_n)$ (for $1 \leqq i \leqq m$) then the completed definition of p is the set of the m 3-valued translations of the clauses, and of the "completed-sequent" of p:

$$?p(x_1, \ldots, x_n) \vdash ?E_1 \vee \ldots \vee E_m$$

If the n-place predicate p does not occur in the head of any program clause, then the completed definition of p is the set containing only the 3-clause: $?p(x_1, \ldots, x_n) \vdash$.

IV.2. Definition: We define **CET** (Clark's equational theory) to be the set of 3-clauses:

(1) $\vdash !x = x$ for each variable x;

(2) $!t(x) = x \vdash$ for each term $t(x)$ different from x in which x occurs;

(3) $!x_1 = y_1, \ldots, !x_n = y_n \vdash !f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$ for each function f;

(4) $!x_1 = y_1, \ldots, !x_n = y_n, \xi p(x_1 \ldots, x_n) \vdash \xi p(y_1, \ldots, y_n)$ for each predicate p;

(5) $!f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n) \vdash !x_i = y_i$ for each n-place function f and for each i ($1 \leqq i \leqq n$);

(6) $!f(x_1, \ldots, x_n) = g(y_1, \ldots, y_m) \vdash$ for all pairs of distinct functions;

(7) $?x = y \vdash !x = y$ for all pairs of variables.

Axioms (1)-(6) are the usual ones; axioms (7) say that the equality relation "$=$" is 2-valued. o

IV.3. Definition: Let **P** be a normal program. The **intuitionistic 3-valued completion** of **P**, denoted **Comp* (P)** is the union of the completed definitions for each predicate p and of **CET**.

IV.4. Proposition: *Let* **P** *be a normal program, then* **Comp* (P)** *is consistent.*

Proof: This is an easy consequence of [15] (theorem 36) because **Comp* (P)** is a three-valued consequence of the three-valued Clark's completion **Comp (P)** used by Kunen (*see* IV.10).

IV.5. DEFINITION (Shepherdson [14]): The notions of **success tree** and **failure tree** of a query **Q** (**Q** being a conjunction of literals) are defined recursively as follows:

Basis:

– if **Q** is "success" ("fail") then the tree consisting of the single node **Q** is a **success** (**failure**) tree for **Q**.

Inductive step:

– if L_i is the chosen literal of **Q**, and if L_i is a positive literal which does not match any clause of **P**, then the tree with a single "fail" node hanging from the root is a **failure tree** for **Q**;

– if L_i is a positive literal which matches one or more clauses of **P**, and if $\mathbf{Q}_1, \ldots, \mathbf{Q}_p$ are the resulting derived queries, then a **success tree** for **Q** is a tree consisting of a success tree for some $\mathbf{Q}_k$ hanging from the root **Q**; a **failure tree** for **Q** is a tree consisting of failure trees for each of $\mathbf{Q}_1, \ldots, \mathbf{Q}_p$ hanging form the root **Q**;

– if L_i is a negative ground literal $\neg A$, a **success tree** for **Q** is a tree consisting of a failure tree for A and a success tree for **Q**′ hanging from the root **Q**, where **Q**′ is the query obtained from **Q** by deleting $\neg A$; a **failure tree** for **Q** is a tree consisting of a failure tree for A and a failure tree for **Q**′ hanging from the root **Q**, or a success tree for A hanging from the root **Q**.

IV.6. LEMMA: (*a*) *If* $p(s_1, \ldots, s_n)$ *and* $p(t_1, \ldots, t_n)$ *are not unifiable then*

$$\mathbf{CET} \vdash_{3\mathrm{I}} ?s_1 = t_1, \ldots, ?s_n = t_n \vdash$$

(*b*) *If* $p(s_1, \ldots, s_n)$ *and* $p(t_1, \ldots, t_n)$ *are unifiable with mgu*

$$\theta = (x_1/r_1, \ldots, x_k/r_k)$$

then

$$\mathbf{CET} \vdash_{3\mathrm{I}} !s_1 = t_1, \ldots, !s_n = t_n \vdash !x_i = r_i \qquad (\textit{for } 1 \leqq i \leqq k)$$

and

$$\mathbf{CET} \vdash_{3\mathrm{I}} !x_1 = r_1, \ldots, !x_k = r_k \vdash !s_i = t_i \qquad (\textit{for } 1 \leqq i \leqq n)$$

Proof: An adaptation of the proof of Lloyd [11] (lemma 15.1).

IV.7. THEOREM: **SLDNF**-*resolution is sound with respect to* **Comp*(P)** *in intuitionistic three-valued logic*, i.e. *suppose that* **Q** *is a conjunction of literals:*

if **Q** *succeeds with answer* θ *then* $\mathbf{Comp^*(P)} \vdash_{3\,\mathrm{I}} \vdash !\,\mathbf{Q}\,\theta$;

if **Q** *fails then* $\mathbf{Comp^*(P)} \vdash_{3\,\mathrm{I}} \vdash !\neg \ni \mathbf{Q}$.

Proof: This theorem and its proof are suggested in [15] (theorem 39) but Shepherdson says that he is "not sure exactly how to formulate that". We use an induction on success and failure trees:

Basis:

(i) **Q** is the positive literal M, and M matches with a clause $\vdash A$, *i.e.* there exists a *mgu* θ of A and M, then since $\vdash !\,A$ is a sequent of **Comp*(P)**, then $\mathbf{Comp^*(P)} \vdash_{3\,\mathrm{I}} \vdash !\,M\,\theta$;

(ii) if $M = p(s_1, \ldots, s_n)$ is the chosen positive literal of **Q**:

– if p does not appear in the head of any clause of program **P**: then the complete definition of p is $?p(x_1, \ldots, x_n) \vdash$; thus obviously $\mathbf{Comp^*(P)} \vdash_{3\,\mathrm{I}} ?M \vdash$ and $\mathbf{Comp^*(P)} \vdash_{3\,\mathrm{I}} \vdash !\neg \ni \mathbf{Q}$;

– if M does not unify with the head of any clause of **P**, then suppose that the 3-valued translations of the clauses with head p are: $!L_{i\,1}, \ldots, !L_{iq_i} \vdash !p(t_{i\,1}, \ldots, t_{in})$; if the completed-sequent of p is: $?p(x_1, \ldots, x_n) \vdash ?E_1 \vee \ldots \vee E_m$ then, by lemma IV.6. $\mathbf{CET} \vdash_{3\,\mathrm{I}} ?(s_1 = t_{i\,1} \,\&\, \ldots \,\&\, s_n = t_{in}) \vdash$ for each i $(1 \leqq i \leqq m)$; hence, $\mathbf{CET} \vdash_{3\,\mathrm{I}} ?s_1 = t_{i\,1} \,\&\, \ldots \,\&\, s_n = t_{in} \,\&\, L_{i\,1} \,\&\, \ldots \,\&\, L_{iq_i} \vdash$ for each i $(1 \leqq i \leqq m)$; and hence, using the completed-sequent of p: $\mathbf{Comp^*(P)} \vdash_{3\,\mathrm{I}} ?p(s_1, \ldots, s_n) \vdash$ and then $\mathbf{Comp^*(P)} \vdash_{3\,\mathrm{I}} \vdash !\neg \ni \mathbf{Q}$.

Inductive step:

– if M_j is the chosen positive literal of **Q** which matches one or more clause of **P**: $M_j = p(s_1, \ldots, s_n)$;

(i) let $\mathbf{Q}_1, \ldots, \mathbf{Q}_p$ be the resulting derived queries and suppose that we have a success tree for one $\mathbf{Q}_i$; then if $\mathbf{Q} = M_1 \,\&\, \ldots \,\&\, M_r$, if ρ is a *mgu* of $p(s_1, \ldots, s_n)$ and of $p(t_{i\,1}, \ldots, t_{in})$ and if

$$\mathbf{Q}_i = M_1 \,\&\, \ldots \,\&\, M_{j-1} \,\&\, L_{i\,1} \,\&\, \ldots \,\&\, L_{iq_i} \,\&\, M_{j+1} \,\&\, \ldots \,\&\, M_r,$$

by induction hypothesis, $\mathbf{Comp^*(P)} \vdash_{3\,\mathrm{I}} \vdash !\,\mathbf{Q}_i\,\sigma$ (if $\mathbf{Q}_i$ succeeds with answer σ); then, using the clause $!L_{i\,1}, \ldots, !L_{iq_i} \vdash !p(t_{i\,1}, \ldots, t_{in})$, we see that $\mathbf{Comp^*(P)} \vdash_{3\,\mathrm{I}} \vdash !\,\mathbf{Q}\,\theta$ (with $\theta = \rho \circ \sigma$);

(ii) let $\mathbf{Q} = M_1 \& \ldots \& M_r$ and $\mathbf{Q}_1, \ldots, \mathbf{Q}_p$ be the resulting derived queries and suppose that we have failure trees for all $\mathbf{Q}_i$; then, by induction hypothesis,

$$\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash ! \neg \ni \mathbf{Q}_i \qquad [\text{for each } i\,(1 \leqq i \leqq p)],$$

and

$$\mathbf{Q}_i = (M_1 \& \ldots \& M_{j-1} \& L_{i\,1} \& \ldots \& L_{iq_i} \& M_{j+1} \& \ldots \& M_r)\rho\,.$$

[if ρ is a *mgu* of $p(s_1, \ldots, s_n)$ and of $p(t_{i\,1}, \ldots, t_{in})$]; then using lemma IV.6 and axioms 7 of **CET**,

$$\mathbf{CET} \vdash_{3\,\mathbf{I}} ?\, s_1 = t_{i\,1} \& \ldots \& s_n = t_{in} \vdash ?\, x_1 = r_1 \& \ldots \& x_k = r_k\,;$$

then, we prove that:

$$\begin{aligned}\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} ?\ni M_1 \& \ldots \& M_{j-1} \& s_1 = t_{i\,1} \& \ldots \& s_n = t_{in}\\ \& L_{i\,1} \& \ldots \& L_{iq_i} \& M_{j+1} \& \ldots \& M_r\\ \vdash ?\ni (M_1 \& \ldots \& M_{j-1} \& L_{i\,1} \& \ldots \& L_{iq_i} \& M_{j+1} \& \ldots \& M_r)\rho\,;\end{aligned}$$

therefore, $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} ! \neg \ni \mathbf{Q}$;

– if M_j is the chosen ground negative literal $\neg A$, and if A has a failure tree and $\mathbf{Q}'$ a success tree, $\mathbf{Q}'$ being the query obtained from $\mathbf{Q}$ by deleting $\neg A$, then by induction hypothesis, $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash ! \mathbf{Q}'\theta$ and $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash ! \mathbf{Q}\theta$;

– if M_j is the chosen ground literal $\neg A$, and if A has a failure tree and $\mathbf{Q}'$ (defined as above) has a failure tree, then by induction hypothesis, $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash ! \neg \ni \mathbf{Q}'$, then $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash ! \neg \ni \mathbf{Q}$;

– if M_j is the chosen ground negative literal $\neg A$, and if A has a success tree, then by induction hypothesis, $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash ! A$ and therefore, $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash ! \neg \ni \mathbf{Q}$.

IV.8. LEMMA: *Let* **P** *be a propositonal normal program and* **Q** *be an atom: if* $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash ! \mathbf{Q}$ *then* **Q** *succeeds under* **SLNDF**-*resolution; if* $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash ! \neg \mathbf{Q}$ *then* **Q** *fails under* **SLNDF**-*resolution.*

Proof: If all clauses of **P** are propositional, then the completed-sequent of a proposition p is $?p \vdash ?E_1 \vee \ldots \vee E_m$ with $E_i = L_{i\,1} \& \ldots \& L_{iq_i}$; we define the **classic-completed** definition of p as the union of the set of the m 3-valued translations of the clauses with head p and of the set of the 3-translated of

N-clauses (*see* section III) obtained from the completed sequent (N-clausal form of the sequent $!\neg E_1 \& \ldots \& \neg E_m \vdash !\neg p$).

Example: Let $r, \neg s \vdash p$ and $t, \neg u \vdash p$ the clauses with head p. Then the classic-completed definition of p is the set of six N-clauses: $!r$, $!\neg s \vdash !p$ and $!t$, $!\neg u \vdash !p$ (the two 3-valued translations) and $!\neg r$, $!\neg t \vdash !\neg p$; $!\neg r$, $!u \vdash !\neg p$; $!s$, $!\neg t \vdash !\neg p$; $!s$, $!u \vdash !\neg p$.

If p does not appear in the head of any program clause, the classic-completed definition of p is $\vdash !\neg p$.

We define $\mathbf{Comp_C(P)}$ as the union of the classic-completed definitions of all propositions p occurring in **P** together with **CET**.

Remark that $\mathbf{Comp_C(P)}$ is a set of 3-translated clauses of N-clauses. Then: if $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash !\mathbf{Q}$, then $\mathbf{Comp_C(P)} \vdash_{3\,\mathbf{K}} \vdash !\mathbf{Q}$ and by results of section III, $\mathbf{Comp_C(P)} \vdash_{3\,\mathbf{I}} \vdash !\mathbf{Q}$; moreover, the deduction only uses the cut and exchange rules (*see* lemma B.1.3); also, if $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash !\neg \mathbf{Q}$, $\mathbf{Comp_C(P)} \vdash_{3\,\mathbf{I}} \vdash !\neg \mathbf{Q}$; we transform this proof to obtain a proof by "hyperresolution": *i.e.* we extend the usual notion of hyperresolution (*see* for example [16]) to formulae $!A$ where A is a literal; then by induction on the proof of $\vdash !\mathbf{Q}$ (or of $\vdash !\neg \mathbf{Q}$), we construct a success (or a failure) tree for **Q**.

IV.9. THEOREM: *If* **P** *is a propositional normal program, then* **SLNDF**-*resolution is complete with respect to* $\mathbf{Comp^*(P)}$ *in three-valued intuitionistic logic* i.e. *if* **Q** *is a conjunction of literals: if* $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash !\mathbf{Q}$ *then* **Q** *succeeds under* **SLNDF**-*resolution; if* $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash !\neg \mathbf{Q}$ *then* **Q** *fails under* **SLNDF**-*resolution.*

Proof: Let $\mathbf{Q} = Q_1 \& \ldots \& Q_p$ (each Q_i being a literal):

– if $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash !\mathbf{Q}$ then, for each i ($1 \leqq i \leqq p$) $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash !Q_i$; then, by lemma IV.8, if Q_i is an atom, Q_i succeeds under **SLNDF**-resolution, and if $Q_i = \neg A_i$, A_i fails under **SLNDF**-resolution. Then, **Q** succeeds under **SLNDF**-resolution;

– if $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash !\neg \mathbf{Q}$, then $\mathbf{Comp_C(P)} \vdash_{3\,\mathbf{K}} \vdash !A_1 \vee \ldots \vee A_p$, with $A_i = \neg Q_i$ if Q_i is an atom, or else (if Q_i is a negative literal) $Q_i = \neg A_i$; by the results of appendix B (lemma B.1.3), $\mathbf{Comp_C(P)} \vdash_{3\,\mathbf{I}} \vdash !A_k$ for one k ($1 \leqq k \leqq p$); hence, $\mathbf{Comp^*(P)} \vdash_{3\,\mathbf{I}} \vdash !A_k$; then, by lemma IV.8, if Q_k is an atom, then Q_k fails and hence, **Q** fails; if $Q_k = \neg A_k$, A_k succeeds, and then Q_k and **Q** fail.

IV.10. Comparison with related work

It is well known that if negation as failure (**SLNDF**-resolution) is sound with respect to Clark's completed **Comp (P)** in classical logic, it is not complete (*see* for example [11] or [15]).

For example, if we consider the program **P** with the unique clause $\neg A \vdash A$ (in Prolog: $A :- \neg A$), then the formula $A \leftrightarrow \neg A$ is an axiom of **Comp (P)** and, since **Comp (P)** is inconsistent, A is a consequence of **Comp (P)** but the query A does not succeed.

Shepherdson ([13] and [14]) shows the soundness of **SLDNF** with respect to Clark's completed **Comp (P)** in intuitionistic logic, but, as the above example shows, we have not the completeness, even for the propositional case and in intuitionistic logic.

Kunen [10] considers a three-valued version of **Comp (P)**: in the completed definition of a predicate $p(x_1, \ldots, x_n) \leftrightarrow E_1 \vee \ldots \vee E_m$, the equivalence takes the value true if the two members take the same value, and takes the value false otherwise. Kunen proves the soundness of *SLDNF* with respect to this classical three-valued logic and the completeness for the propositional case.

Kunen does not define a proof-system for the specific version of three-valued logic that he uses; but we can translate **Comp (P)** into our system, replacing each axiom $p(x_1, \ldots, x_n) \leftrightarrow E_1 \vee \ldots \vee E_m$ by the four sequents:

$$!E_1 \vee \ldots \vee E_m \vdash !p(x_1, \ldots, x_n)$$

$$?E_1 \vee \ldots \vee E_m \vdash ?p(x_1, \ldots, x_n)$$

$$!p(x_1, \ldots, x_n) \vdash !E_1 \vee \ldots \vee E_m$$

$$?p(x_1, \ldots, x_n) \vdash ?E_1 \vee \ldots \vee E_m$$

We obtain a system $\mathbf{Comp_K}$ **(P)** and, since each three-valued structure is a (classical) model of **Comp (P)** iff it is a (classical) three-valued model of $\mathbf{Comp_K}$ **(P)**, and each (classical) three-valued model of $\mathbf{Comp_K}$ **(P)** is an intuitionistic model of **Comp* (P), Comp* (P)** is consistent; moreover, our completeness result is entailed by Kunen's completeness result while our soundness result entails Kunen's soundness result.

APPENDIX A

A. GENTZEN'S SEQUENT CALCULUS AND HORN CLAUSES

In this paragraph we define Gentzen's sequent calculus, and we show some properties of logic programs consisting of Horn clauses, using proof-theoretical methods. Most of these results have been previously proved, by similar or other methods, in [5].

A.1. Gentzen's sequent calculus

In this paper **L** stands for a fixed first-order language; we assume the language **L** has some fixed set of constant, function and relation symbols, and formulae of **L** are defined in the usual way, using $\neg$, &, $\vee$, $\rightarrow$, $\forall$, $\exists$.

We define, as usual, the notions of substitution, closed instance of formula, interpretation, and model of a set of formulae.

A.1.1. DEFINITION: We define the formal system **LK**:

(i) A sequent in **LK** is a formal expression $\Gamma \vdash \Delta$, where Γ and Δ are finite sequences (possibly empty) of formulae in **L**.

(ii) The sequent calculus **LK** is defined as follows:

(1) **axioms**: $A \vdash A$ for each atomic formula A;

(2) **logical rules**:

– **conjunction**

$$\frac{\Gamma \vdash \Delta, A \qquad \Lambda \vdash \Pi, B}{\Gamma, \Lambda \vdash \Delta, \Pi, A \,\&\, B}\,\mathbf{r}\&$$

$$\frac{A, \Gamma \vdash \Delta}{A \,\&\, B, \Gamma \vdash \Delta}\,\mathbf{l1}\& \qquad \frac{B, \Gamma \vdash \Delta}{A \,\&\, B, \Gamma \vdash \Delta}\,\mathbf{l2}\&$$

– **disjunction**

$$\frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, A \vee B}\,\mathbf{r1}\vee \qquad \frac{\Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \vee B}\,\mathbf{r2}\vee$$

$$\frac{A, \Gamma \vdash \Delta \qquad B, \Lambda \vdash \Pi}{A \vee B, \Gamma, \Lambda \vdash \Delta, \Pi}\,\mathbf{l}\vee$$

– **negation**

$$\frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta}\mathbf{l}\neg \qquad \frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A}\mathbf{r}\neg$$

– **implication**

$$\frac{\Gamma \vdash \Delta, A \qquad B, \Lambda \vdash \Pi}{A \to B, \Gamma, \Lambda \vdash \Delta, \Pi}\mathbf{l}\to \qquad \frac{A, \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \to B}\mathbf{r}\to$$

– **for all**

$$\frac{A(t), \Gamma \vdash \Delta}{\forall x, A(x), \Gamma \vdash \Delta}\mathbf{l}\forall(**) \qquad \frac{\Gamma \vdash \Delta, A(x)}{\Gamma \vdash \Delta, \forall x, A(x)}\mathbf{r}\forall(*)$$

– **there is**

$$\frac{A(x), \Gamma \vdash \Delta}{\ni x, A(x), \Gamma \vdash \Delta}\mathbf{l}\ni(*) \qquad \frac{\Gamma \vdash \Delta, A(t)}{\Gamma \vdash \Delta, \ni x, A(x)}\mathbf{r}\ni(**)$$

(*) we have the following restriction on variables: x not free in Γ, Δ.

(**) t is an arbitrary term of **L**.

(3) **structural rules**:

– **weakening**

$$\frac{\Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}\mathbf{l}\mathbf{W} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A}\mathbf{r}\mathbf{W}$$

– **contraction**

$$\frac{A, A, \Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}\mathbf{l}\mathbf{C} \qquad \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A}\mathbf{r}\mathbf{C}$$

– **exchange**

$$\frac{\Pi, A, B, \Gamma \vdash \Delta}{\Pi, B, A, \Gamma \vdash \Delta}\mathbf{l}\mathbf{E} \qquad \frac{\Gamma \vdash \Delta, A, B, \Pi}{\Gamma \vdash \Delta, B, A, \Pi}\mathbf{r}\mathbf{E}$$

(4) **cut**

$$\frac{\Gamma \vdash \Delta, A \qquad A, \Lambda \vdash \Pi}{\Gamma, \Lambda \vdash \Delta, \Pi}\mathbf{Cut}$$

(we say that A is the **cut-formula**).

A.1.2. *Terminology and notations*

(i) We define, as usual the notion of proof in a formal system.

(ii) The rules (1), (2) and (3) are the cut-free rules; a cut-free proof is a proof using only the cut-free rules.

(iii) The logical rules are divided into right rules and left rules according to the fact that the main formula (for example in $r\&$ the main formula is $A \& B$) appears in the right part or in the left part of the sequent.

(iv) If we want to indicate the use of exchange rules, we shall often write $\overline{\overline{\quad}}$ instead of $\overline{\quad}$:

$\dfrac{\Gamma \vdash \Delta}{\overline{\overline{\Gamma' \vdash \Delta'}}}$ means that $\Gamma' \vdash \Delta'$ has been obtained from $\Gamma \vdash \Delta$ by a finite number (possibly zero) of exchange rules; for example,

$\dfrac{\Gamma \vdash \Delta, A \qquad A, \Lambda \vdash \Pi}{\overline{\overline{\Gamma, \Lambda \vdash \Delta, \Pi}}}$ **Cut** means that finitely many exchange rules, together with one application of cut have been used.

A.1.3. DÉFINITIONS:

(i) If **M** is a structure (or interpretation) for **L**, we define the language **L**[**M**] by adding to **L** new constants $\underline{c}$ for all $c \in |\mathbf{M}|$, and we associate, a value $\mathbf{M}(t)$ [resp. $\mathbf{M}(A)$] to each term t (resp. formula A) of **L**[**M**]. We use the notation $\mathbf{M} \models A$ for $\mathbf{M}(A) = t$ (*i.e.* true) and $\mathbf{M} \not\models A$ for $\mathbf{M}(A) = f$ (*i.e.* false).

(ii) The closed sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ of **L**[**M**] is valid in the structure iff:

– if $n \neq 0$ and $m \neq 0$: if $M \models A_1$ and ... and $M \models A_n$ then $M \models B_1$ or ... or $M \models B_m$;

– if $n \neq 0$ and $m = 0$ then $M \not\models A_1$ or ... or $M \not\models A_n$;

– if $n = 0$ and $m \neq 0$ then $M \models B_1$ or ... or $M \models B_n$;

– if $n = 0$ and $m = 0$, the sequent $\vdash$ means absurdity.

A.1.4. DEFINITION: Let **S** be a set of sequents of **LK**. A model **M** of **S** is a structure such that any closed instance of a sequent of **S** is valid in **M**. **S** is consistent iff it has at least one model.

We denote by **SLK** the extension of the calculus **LK** obtained by adding the sequents of **S** to the axioms of **LK**. We say that **SLK** is a theory and that the sequents of **S** are the proper axioms of the theory. If a sequent $\Gamma \vdash \Delta$ is provable in **SLK**, we write $\mathbf{S} \vdash_{\mathbf{K}} \Gamma \vdash \Delta$.

A.1.5. THEOREM (soundness): *The rules of* **LK** *preserve validity of the sequents:* i.e. *if all closed instances of the premisses are valid, all closed instances of the conclusion are valid.*

Thus, if **S** is a set of sequents, and if the sequent $\Gamma \vdash \Delta$ is provable in **SLK**, $\Gamma \vdash \Delta$ is valid in any model of **S**.

A.1.6. THEOREM (completeness): *If all closed instances of a sequent* $\Gamma \vdash \Delta$ *of* **LK** *are valid in all structures for* **L**, *then* $\Gamma \vdash \Delta$ *is provable in* **LK**.

A.1.7. THEOREM: $\Gamma \vdash \Delta$ *enjoys cut-elimination* (i.e. *the cut rule is redundant: if a sequent of* **LK** *is provable, it is cut-free provable*).

A.1.8. *Remark:* The above theorem is Gentzen's famous Hauptsatz. Consequently, we have a subformula property: for any cut-free proof of a sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$, each formula occurring in the proof is a subformula of one A_i or of one B_i.

A.1.9. THEOREM: *Let* **S** *be a set of sequents of* **LK** *closed under substitution* (i.e. *if* $\Gamma(x) \vdash \Delta(x) \in \mathbf{S}$ *then, for each term* t, $\Gamma(t) \vdash \Delta(t) \in \mathbf{S}$).

If a sequent is provable in **SLK**, *then any closed instance of this sequent is valid in any model of* **S** (*soundness*).

If a sequent $\Gamma \vdash \Delta$ *of* **LK** *is valid in any model* **M** *of* **S**, *then it is provable in* **SLK** (*completeness*). *Moreover there is a proof in which all cut-formulae occur in some sequent of* **S**; *hence, any formula occurring in this proof is a subformula of a formula occurring in* **S**, *or in* $\Gamma \vdash \Delta$.

A.2. The intuitionistic case

A.2.1. DEFINITIONS: (i) A sequent of **LI** is a formal expression $\Gamma \vdash \Delta$ such that Γ is a finite sequence (possibly empty) of formulae of **L**, and Δ is empty, or consists in one formula of **L**. We say that $\Gamma \vdash \Delta$ is an intuitionistic sequent.

(ii) The sequent calculus **LI** is the subsystem of **LK** obtained from **LK** by using the following restrictions:

– all sequents considered are intuitionistic sequents;

– we consider as rules of **LI** only the rules of **LK** which, applied to intuionistic sequents, yield intuitionistic sequents [with an exception for the rule (lv)]:

(*a*) for logical rules: the rule (lv) is replaced by $\dfrac{A, \Gamma \vdash \Delta \qquad B, \Lambda \vdash \Delta}{A \vee B, \Gamma, \Lambda \vdash \Delta}$ **lv**

(*b*) the only right structural rules are (*r W*) [because (*r E*) and (*r C*) necessitate more than one formula on the right].

(ii) We define **SLI** from **LI**, as we have defined **SLK** from **LK**. If a sequent $\Gamma \vdash \Delta$ is provable in **SLI**, we write $\mathbf{S} \vdash_{\mathbf{I}} \Gamma \vdash \Delta$.

A.2. Horn clauses

In the following part of this section let **S** be a set of Horn clauses, closed under substitution.

A.3.1. LEMMA (Girard [8]): *If the clause $\Gamma \vdash \Delta$ is provable in* **SLK**, *then there exists a Horn clause $\Gamma' \vdash \Delta'$ provable in* **SLI**, *and any formula A occurring in Γ' is a formula of Γ and if Δ' is not empty, it consists of one formula A of Δ; moreover, no weakening nor contraction rule is used in the proof.*

Proof: By theorem A.1.9, there exists a proof of $\Gamma \vdash \Delta$ where all the cuts are on formulae occurring in a sequent of **S**; first, remark that, by subformula property, no logical rule is used in the proof; we construct, by induction on the proof **D**, a proof $t(\mathbf{D})$ and we verify that the conditions of the theorem are fulfilled:

- if **D** is an axiom we set $t(\mathbf{D}) = \mathbf{D}$;
- if the last rule of **D** is a structural rule:

$$\mathbf{D}: \quad \frac{\begin{array}{c}\mathbf{P}\\ \Gamma_1 \vdash \Delta_1\end{array}}{\Gamma \vdash \Delta}$$

we set $t(\mathbf{D}) = t(\mathbf{P})$;

- if the last rule of **D** is a cut:

$$\mathbf{D}: \quad \frac{\begin{array}{c}\mathbf{P}\\ \Gamma \vdash \Delta, A\end{array} \qquad \begin{array}{c}\mathbf{Q}\\ A, \Lambda \vdash \Pi\end{array}}{\Gamma, \Lambda \vdash \Delta, \Pi}\ \mathbf{Cut}$$

by induction hypothesis, we have proofs $t(\mathbf{P})$ and $t(\mathbf{Q})$ of sequents $\Gamma_1 \vdash \Delta_1$ and $\Lambda_1 \vdash \Pi_1$; if $\Delta_1 \neq A$, we set $t(\mathbf{D}) = t(\mathbf{P})$; if $\Delta_1 = A$ and A does not occur in Λ_1, we set $t(\mathbf{D}) = t(\mathbf{Q})$; if $\Delta_1 = A$ and A occurs n times in Λ_1, then we set:

$$t_1(\mathbf{D}): \quad \frac{\begin{array}{c}t(\mathbf{P})\\ \Gamma_1 \vdash A\end{array} \qquad \begin{array}{c}t(Q)\\ A, \Lambda_2 \vdash \Pi_1\end{array}}{\Gamma_1, \Lambda_2 \vdash \Pi_1}\ \mathbf{Cut}$$

and we apply cuts: using $t(\mathbf{P})$ and $t_i(\mathbf{D})$, we define $t_{i+1}(\mathbf{D})$; finally, $t(\mathbf{D}) = t_n(\mathbf{D})$.

Remark that if a negative clause of **S** has been used, then Δ' is empty, and only a negative clause is used in the proof $t(\mathbf{D})$.

A.3.2. COROLLARY: *We denote by* **SLI°** *the sybsystem of* **SLI**, *having as only rules the exchange and cut rules. Let* **S** *be a consistent set of Horn clauses:*

(i) *if the closure of a positive clause* $\vdash A$ *is valid in any model of* **S**, *then* $\vdash A$ *is provable in* **SLI°**;

(ii) *if the closure of a negative clause* $A \vdash$ *is valid in any model of* **S**, *then* $\Gamma \vdash$, Γ *being a sequence containing only occurrences of* A, *is provable in* **SLI°**.

Proof: (i) By theorem A.1.9 and lemma A.3.1 the sequent $\vdash$ or the sequent $\vdash A$ are provable in **SLI**. Since **S** is consistent, $\vdash$ is not provable (soundness of **SLK** and thus, of **SLI**); hence, we obtain the result, since if an axiom $A \vdash A$ is used, the only rule that we can apply is a cut, therefore we can suppress it;

(ii) it is an immediate consequence of A.1.9 and A.3.1.

A.3.3. LEMMA: *The lemma* A.3.1 *remains true if* Δ *is a sequence of formulae built with* &, $\vee$, $\ni$.

Proof: By the subformula property, the only logical rules possibly used are: $(r1\vee)$, $(r2\vee)$, $(r\&)$, $(r\ni)$; so, the proof of A.1.2 has to be completed:

– if the last rule is $(r1\vee)$:

$$\mathbf{D}:\quad \frac{\begin{matrix}\mathbf{P}\\ \Gamma \vdash \Delta_1, A\end{matrix}}{\Gamma \vdash \Delta_1, A \vee B}\, r1\vee$$

by induction hypothesis, we have a deduction $t(\mathbf{P})$ of a sequent $\Gamma' \vdash \Delta'_1$; if Δ'_1 is not A, we set $t(\mathbf{D}) = t(\mathbf{P})$; otherwise, we set:

$$t(\mathbf{D}):\quad \frac{\begin{matrix}t(\mathbf{P})\\ \Gamma' \vdash A\end{matrix}}{\Gamma' \vdash A \vee B}\, \mathbf{r1}\vee$$

– if the last rule is $(r2\vee)$, the proof is similar;

– if the last rule is $(r\&)$

$$\mathbf{D}:\quad \frac{\begin{matrix}\mathbf{P}\\ \Gamma_1 \vdash \Delta_1, A\end{matrix} \qquad \begin{matrix}\mathbf{Q}\\ \Lambda_1 \vdash \Pi_1, B\end{matrix}}{\Gamma_1, \Lambda_1 \vdash \Delta_1, \Pi_1, A \,\&\, B}\, \mathbf{r}\&$$

by induction hypothesis, we have proofs $t(\mathbf{P})$ and $t(\mathbf{Q})$ of sequents $\Gamma'_1 \vdash \Delta'_1$ and $\Lambda'_1 \vdash \Pi'_1$; if Δ'_1 is not A, we set $t(\mathbf{D}) = t(\mathbf{P})$; if Δ'_1 is not B, we set $t(\mathbf{D}) = t(\mathbf{Q})$; otherwise we set:

$$t(\mathbf{D}):\quad \frac{\begin{array}{c} t(\mathbf{P}) \\ \Gamma'_1 \vdash A \end{array} \qquad \begin{array}{c} t(\mathbf{Q}) \\ \Lambda'_1 \vdash B \end{array}}{\Gamma'_1, \Lambda'_1 \vdash A \,\&\, B}\, r\&$$

– if the last rule is $(r\exists)$

$$\mathbf{D}:\quad \frac{\begin{array}{c} \mathbf{P} \\ \Gamma_1 \vdash \Delta_1, A(t) \end{array}}{\Gamma_1 \vdash \Delta_1, \exists x, A(x)}\, \mathbf{r}\exists$$

by induction hypothesis, we have a proof $t(P)$ of a sequent $\Gamma'_1 \vdash \Delta'_1$; if Δ'_1 is not $A(t)$, we set $t(\mathbf{D}) = t(\mathbf{P})$; otherwise we set:

$$t(\mathbf{D}):\quad \frac{\begin{array}{c} t(\mathbf{P}) \\ \Gamma'_1 \vdash A(t) \end{array}}{\Gamma'_1 \vdash \exists x, A(x)}\, \mathbf{r}\exists$$

A.3.4. COROLLARY: *Let* **S** *be a consistent set of Horn clauses. If any closed instance of a formula* $\exists x_1 \ldots \exists x_q, A_1 \vee \ldots \vee A_n$, *each* A_i *being a conjunction of atoms,* $(A_i = B_{i1} \,\&\, \ldots \,\&\, B_{il_i})$ *is valid in any model of* **S**, *then there exists an index* i $(1 \leqq i \leqq n)$ *and terms* $t_1, \ldots, t_q$ *such that* $B_{ik}(t_1, \ldots, t_q)$ *(for each* k, $1 \leqq k \leqq l_i$*) is provable in* **SLI°**.

Proof: It is a consequence of properties of intuitionistic calculus: by theorem A.1.9 and lemma A.1.4 the sequent $\vdash \exists x_1 \ldots \exists x_q, A_1 \vee \ldots \vee A_n$ is provable in **SLI** (without using contractions nor weakenings), then by induction on the proof, we construct the required proofs: remark first that, by the subformula property, all rules $(r\&)$ are applied before $(r1\vee)$ or $(r2\vee)$, and that these ones are applied before $(r\exists)$ rules; then, by induction on the proof we may delete all $(r\&)$, $(r1\vee)$, $(r2\vee)$, $(r\exists)$ rules; and we obtain the proofs.

A.3.5. *Remark:* The above corollary is not true if one of the B_{ik}'s is a negative literal; take, for example for **S**, the Horn clause: $p(a), p(b) \vdash$; then $\vdash \exists x \neg p(x)$ is valid in any model of **S**, and thus, is provable in **SLK**, but not in **SLI**; so, there does not exist a term t such that $p(t)$ is provable in **SLI**

(neither in **SLK**). A proof in **SLK** of $\vdash \ni x \neg p(x)$ is the following:

$$
\begin{array}{l}
p(a), p(b) \vdash \\
\hline{\scriptstyle}\ \mathbf{r}\neg \\
p(b) \vdash \neg p(a) \\
\hline{\scriptstyle}\ \mathbf{r}\neg \\
\vdash \neg p(a), \neg p(b) \\
\hline{\scriptstyle}\ \mathbf{r}\ni \\
\vdash \neg p(a), \ni x \neg p(x) \\
\hline\hline{\scriptstyle}\ \mathbf{r}\ni \\
\vdash \ni x \neg p(x), \ni x \neg p(x) \\
\hline{\scriptstyle}\ \mathbf{r}\mathrm{C} \\
\vdash \ni x \neg p(x)
\end{array}
$$

A.3.6. LEMMA: *Let* **S** *be a consistent set of Horn clauses. If any closed instance of a formula* $\ni x_1 \ldots \ni x_q$, Q *(with* $Q = A_1 \vee \ldots \vee A_n$ *each* A_i *being an anti-Horn clause:* see *definition* II.1) *is valid in any model of* **S**, *then there exists tuples of terms* $(t_{11}, \ldots, t_{1q}), \ldots, (t_{k1}, \ldots, t_{kq})$ *satisfying the following property:*

$$\vdash Q(t_{11}/x_1, \ldots, t_{1q}/x_q), \ldots, Q(t_{k1}/x_1, \ldots, t_{kq}/x_q)$$

is provable in **SLK**.

Proof: By theorem A.1.9, the sequent $\vdash \ni x_1 \ldots \ni x_q$, Q is provable in **SLK** and we can choose the proof such that any cut-formula occurs in a sequent of **S**. Observe that, by the subformula property, only $(r\neg)$, $(r\&)$, $(r1\vee)$, $(r2\vee)$, $(r\ni)$, are used as logical rules in the proof and that $(r\neg)$ rules are used, for any right occurrence of a formula, before $(r\&)$, $(r\&)$ are used before $(r1\vee)$, $(r2\vee)$ rules and the latter before $(r\ni)$-rules; then delete in the proof all applications of a $(r\ni)$-rule and weakenings or contractions so that the main formula is existential; we obtain a proof of a sequent

$$\vdash Q(t_{11}/x_1, \ldots, t_{1q}/x_q), \ldots, Q(t_{k1}/x_1, \ldots, t_{kq}/x_q).$$

Then, by soundness of **SLK**, the closure of the formula

$$Q(t_{11}/x_1, \ldots, t_{1q}/x_q) \vee \ldots \vee Q(t_{k1}/x_1, \ldots, t_{kq}/x_q)$$

is valid in any model of **S**.

A.4. SLD-resolution

In this section, we use **SLD**-resolution. We suppose that the reader is familiar with this notion. If not, he can consult [11] or [5]. We define in the same way as Gallier the notion of logic program (*see* II.2).

A.4.1. THEOREM (completeness): *Let* (**P**, **Q**) *be a logic program; then,*

(1) *either* **P** *is inconsistent, and there exists at least one* **SLD**-*refutation of* **P** *with goal a negative clause of* **P**; *no other negative clause is used in the refutation;*

(2) *or* **P** *is consistent and then,*

(i) *if* $\mathbf{Q} = A_1 \vee \ldots \vee A_n$, *each* A_i *being a conjunction of atoms* B_{ij}: *if the query* $\ni \mathbf{Q}$ *is a logical consequence of* **P**, *then it is an intuitionistic consequence of* **P** *and there exists at least one n-uple* $(t_1, \ldots, t_q)$ *of terms of* **L** *satisfying the property that* $\mathbf{Q}(t_1/x_1, \ldots, t_q/x_q)$ *is a (intuitionistic) logical consequence of* **P**; *for any such n-uple there exists one* **SLD**-*refutation with answer substitution* θ *and a ground substitution* ρ *with the restriction of* $\theta \circ \rho$ *to* $(x_1, \ldots, x_q)$ *being* $(t_1/x_1, \ldots, t_q/x_q)$

(ii) *if* $\mathbf{Q} = A_1 \vee \ldots \vee A_n$ *is a disjunction of anti-Horn clauses, then if the query* $\ni \mathbf{Q}$ *is a logical consequence of* **P**, *there exists a sequence,* $(t_{11}, \ldots, t_{1q}), \ldots, (t_{k1}, \ldots, t_{kq})$ *of n-tuples of terms of* **L** *satisfying:*

$$\mathbf{Q}(t_{11}/x_1, \ldots, t_{1q}/x_q) \vee \ldots \vee \mathbf{Q}(t_1/x_{11}, \ldots, t_{kq}/x_q)$$

is a (classical) consequence of **P**; *moreover, there exists a* **SLD**-*refutation and substitutions* $\theta_1, \ldots, \theta_k$ *and a ground substitution* ρ *with the restrictions of* $\theta_1 \circ \rho, \ldots, \theta_k \circ \rho$ *to* $(x_1, \ldots, x_q)$ *being respectively*

$$(t_{11}/x_1, \ldots, t_{1q}/x_q), \ldots, (t_{k1}/x_1, \ldots, t_{kq}/x_q).$$

Proof: If (**P**, **Q**) is a logic program, let **S** be the set (in general infinite) of all clauses obtained by substitution from clauses of **P**. If we consider a proof in **SLI°** (or in **SLI**), only a finite number of clauses of **S** occur in the proof and these clauses are in the form $C\theta$, C being a clause of **P**, and θ a substitution. Remark that a proof of the sequent $\vdash$ in **SLI°** can be transformed into a **SLD**-refutation with goal, the only negative clause used in the proof: we prove this fact for propositional calculus, then we extend it to first-order logic using lifting techniques (this tool is well-known and we don't develop it here); thus,

(1) if **P** is inconsistent we get the result by lemma A.3.1 and the above remark;

(2) if **P** is consistent;

(i) by corollary A.3.4, there is an index i so that for each j $(1 \leqq j \leqq l_i)$ $B_{ij}(t_1/x_1, \ldots, t_q/x_q)$ is provable in **SLI°** (if we suppose that

$$A_i = B_{i1} \& \ldots \& B_{il_i})$$

and only definite clauses are used in the proofs; then, if we add to **S** the closure under substitution of the negative clause $B_{i1}, \ldots, B_{il_i} \vdash$, we obtain a set **S'** and we easily construct a proof in **S' LI°** of the sequent $\vdash$; this proof can be transformed into a **SLD**-refutation satisfying the conditions of the theorem;

(ii) by lemma A.3.6, there exists a proof in **SLK** of the sequent

$$\vdash \mathbf{Q}(t_{11}/x_1, \ldots, t_{1q}/x_q), \ldots, \mathbf{Q}(t_{k1}/x_1, \ldots, t_{kq}/x_q).$$

If we suppose that we add to **S** all substitutions of clauses obtained from negations of formuale A_i $(1 \leqq i \leqq n)$, we obtain a set **S'** of Horn clauses and we construct easily a proof in **S' LK** of the sequent $\vdash$. But, since all clauses in **S'** are Horn clauses, we have a proof in **S' LI°** of the sequent $\vdash$ (by lemma A.3.1), and therefore a **SLD**-refutation.

APPENDIX B

B. THREE-VALUED SEQUENT CALCULUS

B.1. Classical and intuitionistic three valued sequent calculus

B.1.1. Definitions: We suppose that the language **3L** is defined as in III.1.3. We define the formal system **3LK** (we use notations of section III and ξ, η vary through !, ?):

(1) **axioms:** $\xi A \vdash \xi A$ and $! A \vdash ? A$ for each atomic formula A;

(2) **logical rules:**

– **conjunction**

$$\frac{\Gamma \vdash \Delta, \xi A \qquad \Lambda \vdash \Pi, \xi B}{\Gamma, \Lambda \vdash \Delta, \Pi, \xi A \& B}\ \mathbf{r}\&$$

$$\frac{\xi A, \Gamma \vdash \Delta}{\xi A \& B, \Gamma \vdash \Delta}\ \mathbf{l1}\& \qquad \frac{\xi B, \Gamma \vdash \Delta}{\xi A \& B, \Gamma \vdash \Delta}\ \mathbf{l2}\&$$

– **disjunction**

$$\frac{\Gamma \vdash \Delta, \xi A}{\Gamma \vdash \Delta, \xi A \vee B}\,\mathbf{r1}\vee \qquad \frac{\Gamma \vdash \Delta, \xi B}{\Gamma \vdash \Delta, \xi A \vee B}\,\mathbf{r2}\vee$$

$$\frac{\xi A, \Gamma \vdash \Delta \qquad \xi B, \Lambda \vdash \Pi}{\xi A \vee B, \Gamma, \Lambda \vdash \Delta, \Pi}\,\mathbf{l}\vee$$

– **negation**

$$\frac{\Gamma \vdash \Delta, \xi A}{\xi^\circ \neg A, \Gamma \vdash \Delta}\,\mathbf{l}\neg \qquad \frac{\xi A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \xi^\circ \neg A}\,\mathbf{r}\neg$$

– **implication**

$$\frac{\Gamma \vdash \Delta, \xi^\circ A \qquad \xi B, \Lambda \vdash \Pi}{\xi A \to B, \Gamma, \Lambda \vdash \Delta, \Pi}\,\mathbf{l}\to \qquad \frac{\xi^\circ A, \Gamma \vdash \Delta, \xi B}{\Gamma \vdash \Delta, \xi A \to B}\,\mathbf{r}\to$$

– **for all**

$$\frac{\xi A(t), \Gamma \vdash \Delta}{\xi \forall x, A(x), \Gamma \vdash \Delta}\,\mathbf{l}\forall(**) \qquad \frac{\Gamma \vdash \Delta, \xi A(x)}{\Gamma \vdash \Delta, \xi \forall x, A(x)}\,\mathbf{r}\forall\,(*)$$

– **there is**

$$\frac{\xi A(x), \Gamma \vdash \Delta}{\xi\, x, A(x), \Gamma \vdash \Delta}\,\mathbf{l}\ni(*) \qquad \frac{\Gamma \vdash \Delta, \xi A(t)}{\Gamma \vdash \Delta, \xi \ni x, A(x)}\,\mathbf{r}\ni(**)$$

(*) we have the following restriction on variables: x not free in Γ, Δ.
(**) t is an arbitrary term of **L**.

(3) **structural rules**

– **weakening**

$$\frac{\Gamma \vdash \Delta}{\xi A, \Gamma \vdash \Delta}\,\mathbf{l}\mathbf{W} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \xi A}\,\mathbf{r}\mathbf{W}$$

– **contraction**

$$\frac{\xi A, \xi A, \Gamma \vdash \Delta}{\xi A, \Gamma \vdash \Delta}\,lC \qquad \frac{\Gamma \vdash \Delta, \xi A, \xi A}{\Gamma \vdash \Delta, \xi A}\,rC$$

– **exchange**

$$\frac{\Pi, \xi A, \xi B, \Gamma \vdash \Delta}{\Pi, \xi B, \xi A, \Gamma \vdash \Delta}\,\mathbf{lE} \qquad \frac{\Gamma \vdash \Delta, \xi A, \xi B, \Pi}{\Gamma \vdash \Delta, \xi B, \xi A, \Pi}\,\mathbf{rE}$$

(4) **cut**

$$\frac{\Gamma \vdash \Delta, \xi A \qquad \xi A, \Lambda \vdash \Pi}{\Gamma, \Lambda \vdash \Delta, \pi}\,\mathbf{Cut}$$

We define the intuitionistic three-valued sequent calculus **3LI** from **3LK**, just as we have defined the sequent calculus **LI** from **LK**.

B.1.2. DEFINITIONS AND THEOREMS: We define classical three-valued structures and models (*see* III.1.4). For intuitionistic three-valued models, *see* [7].

Theorems A.1.5, A.1.6, A.1.9 (soundness, completeness, cut-elimination) can be extended to **3LK** (and **3LI**).

B.1.3. LEMMA: *Let* **S*** *be a consistent set of* 3-*translated of Horn N-clauses of a set* **S** (see *definition* III.4.2); *if* $\mathbf{S}^* \vdash_{\mathbf{3K}} \vdash ! A_1 \vee \ldots \vee A_p$ *each* A_i *being a literal, then* $\mathbf{S}^* \vdash_{\mathbf{3I}} \vdash ! A_k$ *for one* k $(1 \leqq k \leqq p)$.

Proof: It is a consequence of lemma A.3.4: replace (as in III.4.2) in the proof (in **S*3LK**) of $\vdash ! A_1 \vee \ldots \vee A_p$, each literal $\neg p(t_1, \ldots, t_n)$ by $p^*(t_1, \ldots, t_n)$ and remove all symbols "!": we obtain a proof in **SLK**; apply corollary A.3.4 to obtain a proof in **SLI** of $\vdash A_k$ or of $\vdash B_k^*$ (with $A_k = \neg B_k$) (for one k such that $1 \leqq k \leqq p$); then replace in the proof each atom $p^*(t_1, \ldots, t_n)$ by $!\neg p(t_1, \ldots, t_n)$ and each atom $p(t_1, \ldots, t_n)$ by $!p(t_1, \ldots, t_n)$: we obtain the required proof.

REFERENCES

1. S. CERRITO, Negation as Failure: a Linear Axiomatization, *Rapport de recherche*, L.R.I., Orsay, 1988.
2. K. L. CLARK, Negation as Failure, in Logic and Databases, H. GALLAIRE and J. MINER Eds., *Plenum Press*, New York, 1978, pp. 293-322.
3. J. P. DELAHAYE, Chaînage avant et calcul de modèles booléens et trivalués, *Expert systems and their applications*, Avignon, 1987.
4. M. FITTING, A Kripke-Kleene Semantics for General Logic programs, *J. Logic Programming*, 1985, *2*, pp. 295-321.
5. J. GALLIER, Logic for Computer Science: Foundations of Automatic Theorem proving, *Harper and Row*, New York, 1986.
6. G. GENTZEN, Collected works, SZABO Ed., *North Holland*, Amsterdam, 1969.

7. J. Y. GIRARD, Three-Valued Logic and Cut-Elimination: The Actual Meaning of Takeuti's Conjecture, *Dissertationes Math.*, Warszawa, 1976.
8. J. Y. GIRARD, Théorie de la démonstration, *Cours de* 3° *cycle*, Université Paris-VII, 1986.
9. J. Y. GIRARD, Proof Theory and Logical Complexity, 1987, *1*, Napoli, Bibliopolis.
10. K. KUNEN, Negation in logic programming, *J. Logic programming*, 1987, *4*, pp. 289-308.
11. J. W. LLOYD, Foundations of Logic programming, Second Edition, *Springer Verlag*, Berlin, 1987.
12. P. H. SCHMITT, Computational Aspects of Three Valued Logic, *Proc. 8th Conf. Automated Deduction, Lecture Notes in Comput. Sci.*, 1986, *230*, Springer-Verlag, pp. 190-198.
13. J. C. SHEPHERDSON, Negation as Failure: A Comparison of Clark's Completed Data Base and Reiter's Closes World Assumption, *J. Logic Programming*, 1984, *1*, pp. 51-81.
14. J. C. SHEPHERDSON, Negation as Failure II, *J. Logic programming*, 1985, *3*, pp. 185-202.
15. J. C. SHEPHERDSON, Negation in Logic Programming, in J. MINKER Éd., Foundations of Deductive Databases and Logic Programming, *Morgan Kaufman*, Los Altos, 1988, pp. 19-88.
16. M. H. VAN EMDEN and R. KOWALSKI, The Semantics of Predicate Logic as a Programming Language, *J. Assoc. Comput. Mach.*, 1976, *23*, pp. 733-742.