Towards Abductive Reasoning in First-order Logic

A. LILIANA REYES-CABELLO, *Departamento de Matemáticas. Facultad de Ciencias. Universidad Nacional Autónoma de México. Coyacan, México, D.F. (Mexico). Email: liliana@ciencias.unam.mx*

ATOCHA ALISEDA-LLERA, *Instituto de Investigaciones Filosóficas. Universidad Nacional Autónoma de México. México D.F. (Mexico). Email: atocha@filosoficas.unam.mx*

ÁNGEL NEPOMUCENO-FERNÁNDEZ, *Unidad de Lógica, Lenguaje e Información. Dpto. de Filosofía y Lógica. Universidad de Sevilla. Sevilla (Spain). Email: nepomuce@us.es*

Abstract

Abductive problems have been widely studied in propositional logic. First order abduction, however, has been viewed as intractable, for the undecidability of logical consequence. In this paper, we propose a notion of abductive problem, N-abductive problem, which is relative to the cardinality of the minimal model satisfying the given theory. We use a notion of restricted satisfaction, also relative to a domain cardinality. Finally, we propose an effective procedure for the searching of abductive solutions, by means of a modification of Beth's tableaux.

Keywords: abduction, semantic tableaux.

1 Introduction

Roughly speaking, in a logical framework, abductive reasoning consists in inferring an explanation α from premises $\alpha \rightarrow \varphi$ and φ. This rule in an incorrect logical one, but it nevertheless reflects some form of human reasoning, as that found in diagnosis and common sense reasoning, where the search of possible causes for a given event is frequently involved.

Abduction has been studied very much at a propositional level, witnessed by the many theoretical and practical approaches found in the literature. There are algorithms and implementations which aim at finding solutions to propositional abductive problems[1, 2, 12, 8]. But once me move to first order logic ($\mathcal{FOL}$), the field is not quite explored. The reason is that the notion of an abductive problem is generally characterized in terms of logical consequence, and this notion is undecidable in $\mathcal{FOL}$. Therefore, in a first order language, abduction is regarded as intractable.

However, the undecidability of $\mathcal{FOL}$ has not been an obstacle to tackle problems involving logical consequence from a computational view, for there are decidable fragments of $\mathcal{FOL}$. The existence of automated provers is an evidence of this claim. These automated provers are by no means trivial, neither in their implementation nor in their applications, and are able to solve the problems involving logical consequence in many fragments of $\mathcal{FOL}$. A good

doi:10.1093/jigpal/jzk019

example is OTTER.[1]

Following the suggestions coming from these systems, one may ask the following question:

Are there sets of formulae (theories and facts) where it is possible to tackle the problem of first order abduction?

A preliminary study, in which the notions involved in first order abduction are analyzed, appears in [8]. In this work, a theoretical method to obtain solutions is proposed, by means of Beth's semantic tableaux and sequent calculus.

In this paper we tackle the problem of first order abduction given the theory and the conclusion as formulae in prenex normal form, without equality or function symbols. We will study the class of abductive problems $\langle \Theta, \varphi \rangle$ such that the Θ is satisfiable by a finite model. To this aim we introduce the notion of *n-abductive problem*, relative to the cardinality of those minimal models which satisfy Θ. The characterization of an n-abductive problem is based in turn on the notions of n-satisfaction and logical n-consequence, as proposed in [9].

We introduce a modification of Beth's tableaux, which we call N-tableaux, in order to have an effective procedure to obtain solutions for n-abductive problems. Soundness and completeness of N-tableaux are proved, with respect to logical n-consequence.

Overview

We end the introduction with a short description of the contents of this paper. In the second section we present the basics of the syntax and semantics of the particular first order language we use, as well as the notion of n-satisfaction, logical n-consequence, and the standard definition of abductive problem. We limit the question of first order abduction to sets of decidable and satisfiable formulae in finite domains. In the third section, we define n-abductive problems in terms of the notion of n-satisfaction. We then characterize the solutions of these problem by means of the concept of logical n-consequence. In the next section, we propose a modification of the tableau method, the N-tableaux, in order to obtain an effective procedure for the searching of solutions to n-abductive problems. We prove soundness of N-tableaux with respect to the notion of n-satisfaction. In the fifth section we characterize an abductive problem by means of n-tableaux. Finally, we present our conclusions and open lines for future research.

2 Preliminaries

Abduction is a form of logical reasoning different from deduction and induction, because in this type of reasoning the intended inference is not a valid conclusion, but rather an explanation from a given theory and a fact.

Definition 2.1 (abductive problem)
Let Θ be a finite set of formulae and φ a formula of $\mathscr{L}$, such that $\Theta \not\models \varphi$ and $\Theta \not\models \neg\varphi$. Hence, an explanation of φ is required. We say then that Θ and φ form an abductive problem, denoted by $\langle \Theta, \varphi \rangle$.

A solution for an abductive problem is any formula α such that together with the theory Θ, entails fact φ. Note that the solution for an abductive problem may not be unique. The

[1] See `http://www-unix.mcs.anl.gov/AR/otter`

set of all possible solutions is defined, in formal terms, as follows:

Definition 2.2 (solution set)
Let $\langle\Theta,\varphi\rangle$ be an abductive problem; we define the solution set of $\langle\Theta,\varphi\rangle$, in symbols $Sol(\langle\Theta,\varphi\rangle)$, in the following way:

$$Sol(\langle\Theta,\varphi\rangle) = \{\psi \in \mathscr{L} \mid \Theta \cup \{\psi\} \models \varphi\}$$

We should mention that the set $Sol(\langle\Theta,\varphi\rangle)$ contains every possible solution, including the "undesirable" ones, which are, in the least, the trivial ones. That is, those that include φ or that are inconsistent with the theory. It is clear that such formulae, though solutions in semantical strict sense, are not explanations for a fact φ. In [1] four types of abductive solutions are proposed and classified according to the conditions they verify. We will center our attention in three of them: inference, consistency, and being explanatory. Formally,

Definition 2.3 (solution for an abductive problem)
Let $\langle\Theta,\varphi\rangle$ be an abductive problem and α a formula of $\mathscr{L}$; α is a solution or explanation for $\langle\Theta,\varphi\rangle$ if it verifies the following conditions:

1. $\alpha \in Sol(\langle\Theta,\varphi\rangle)$, that is, $\Theta \cup \{\alpha\} \models \varphi$.
2. $\Theta \cup \{\alpha\}$ is consistent ($\Theta \cup \{\alpha\} \not\models \bot$).
3. $\alpha \not\models \varphi$.

Observation 2.4
$\alpha \not\models \neg\varphi$ is a consequence of the above conditions. If $\alpha \models \neg\varphi$ then $\Theta \cup \{\alpha\} \models \neg\varphi$ and by (1) we have that $\Theta \cup \{\alpha\} \models \varphi$. But $\Theta \cup \{\varphi\}$ is inconsistent, contradicting 2. So, $\alpha \not\models \neg\varphi$.

The results proposed in this paper are valid for sets of formulae of first order languages without equality nor function symbols.

Let $\mathscr{L}$ be a first order formal language, defined over the following full alphabet:

- An enumerable infinite set of variables $Var = \{x_1, x_2, \ldots\}$.
- Logical constants: $\bot, \top$.
- Logical operators or connectives: $\neg, \wedge, \vee$.[2]
- Quantifiers: $\exists, \forall$.
- Auxiliary symbols: '(', ')', ','.

The signature of the language is given by:

- A non-empty set of predicate symbols, which is a subset of $Pred = \{P_m^n \mid n, m \in \mathbb{N}, n \geq 1\}$, where each $n \geq 1$ (called the *index* of *arity* of the symbol P_m), denotes the number of arguments of the predicate P_m^n.
- A set of constants, which is a subset of $Cons = \{c_1, c_2, \ldots\}$

In practice, we will use metavariables $P, Q, R, ...$ to denote predicate symbols.

Definition 2.5 (formula complexity)
The complexity of a formula φ is the number of connectives and quantifiers which appear in φ.

[2] Any formula can be represented by an equivalent one, containing at most these three logical connectives, for they constitute a complete set of connectives.

Regarding the semantics, we will use the notions of satisfaction and logical consequence, relative to the cardinality of the domain of discourse, introduced in [9] and presented in what follows.

Definition 2.6 (n-satisfaction)
Let φ be a formula of $\mathscr{L}$; we say φ is n-satisfiable if there is a $\mathscr{L}$-structure $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ such that $|\mathcal{D}| = n$ with $n \geq 1$ and $\mathcal{M} \models \varphi$. In symbols, $\mathcal{M} \models_n \varphi$.

Definition 2.7 (Minimal n-satisfiable set of formulae)
A finite set of formulae Γ is minimal n-satisfiable if Γ has a model of cardinality n, and there is no model of Γ of lesser cardinality.

The following result and a proof-sketch were published in [10]. We now present it with a detailed proof. First we introduce a new concept, that of c-equivalence:

Definition 2.8 ($\mathscr{L}$-structures c-equivalence)
Let $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ and $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ be two $\mathscr{L}$-structures and c a constant of $\mathscr{L}$. We say that $\mathcal{M}$ and $\mathcal{M}'$ are c-equivalent if $\mathcal{D} = \mathcal{D}'$ and the interpretation functions $\mathcal{I}$ and $\mathcal{I}'$ differ in at most the constant c. That is, $\mathcal{I} \restriction_{\mathscr{L} \setminus \{c\}} = \mathcal{I}' \restriction_{\mathscr{L} \setminus \{c\}}$. In symbols, $\mathcal{M} =_c \mathcal{M}'$.

Theorem 2.9
If a formula φ is n-satisfiable, for $n \geq 1$, then φ is m-satisfiable for any $m \geq n$.

Proof. Let $\mathcal{D}$ be a universe of discourse with cardinality $n \geq 1$ and $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ an $\mathscr{L}$-structure such that $\mathcal{M} \models_n \varphi$. Let $\mathcal{D}^*$ be a domain with $|\mathcal{D}^*| = m \geq n$, and $\mathcal{D}' \subseteq \mathcal{D}^*$ such that $|\mathcal{D}'| = n$. Let $f : \mathcal{D}' \to \mathcal{D}$ be a bijective function. We define the function $g : \mathcal{D}^* \to \mathcal{D}$, in the following way:

1. If $r \in \mathcal{D}'$, then $g(r) = f(r)$.
2. Else, $g(r) = s$ (s is a distinguished element of $\mathcal{D}$).

Then, we define $\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$ such that $\mathcal{I}^*$ verifies the following conditions:
Let c be a constant of $\mathscr{L}$, then,

$$\mathcal{I}^*(c) = d' \in \mathcal{D}' \;\text{ if } \mathcal{I}(c) = g(d')$$

Observation 2.10
$\mathcal{I}(c) = g(\mathcal{I}^*(c))$ for each $c \in Cons$.

Moreover, given $(s_1, \ldots, s_k) \in \mathcal{D}^{*k}$ and $P^k_m \in Pred$, we define $\mathcal{I}^*(P^k_m)$ as follows:

$$(s_1, \ldots, s_k) \in \mathcal{I}^*(P^k_n) \text{ syss } (g(s_1), \ldots, g(s_k)) \in \mathcal{I}(P^k_m).$$

Once we define $\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$, we can prove that $\mathcal{M} \models_n \varphi$ iff $\mathcal{M}^* \models_m \varphi$. But we will prove the following result, beforehand:

Lemma 2.11
Let $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ and $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ be $\mathscr{L}$-structures and c a constant of $\mathscr{L}$. Let $\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$ and $\mathcal{M}'^* = \langle \mathcal{D}'^*, \mathcal{I}'^* \rangle$ be $\mathscr{L}$-structures defined from $\mathcal{M}$ y $\mathcal{M}'$, respectively, constructed in the described way. Then,

$$\mathcal{M} =_c \mathcal{M}' \text{ syss } \mathcal{M}^* =_c \mathcal{M}'^*.$$

Proof. If $\mathcal{M} =_c \mathcal{M}'$, by definition we have that

$$\mathcal{I}(c') = \mathcal{I}'(c') \text{ for every } c' \neq c.$$

And by observation 2.10

$$g(\mathcal{I}^*(c')) = g(\mathcal{I}^{'*}(c')) \text{ for every } c' \neq c.$$

Remember that $\mathcal{I}^*(c') \in D'$ and $\mathcal{I}^*(c') \in D$, in addition to $g \restriction_{\mathcal{D}'} = f$, being f a bijective function. Therefore,

$$\mathcal{I}^*(c') = \mathcal{I}^{'*}(c')$$

Hence, $\mathcal{M}^{'*} =_c \mathcal{M}^*$. ∎

We now go back to the proof of Theorem 2.9. We perform induction on the construction of φ.

- $\varphi = P_m^k(c_1, \ldots, c_k)$.
  $$\begin{array}{lll} \mathcal{M} \models \varphi & \text{iff} & (\mathcal{I}(c_1), \ldots, \mathcal{I}(c_k)) \in \mathcal{I}(P_m^k) \\ & \text{iff} & (g(\mathcal{I}^*(c_1)), \ldots, g(\mathcal{I}^*(c_k))) \in \mathcal{I}(P_m^k) \\ & \text{iff} & (\mathcal{I}^*(c_1), \ldots, \mathcal{I}^*(c_k)) \in \mathcal{I}^*(P_m^k) \\ & \text{iff} & \mathcal{M}^* \models P_m^k(c_1, \ldots, c_k) \end{array}$$
- For the logical connectives, the proof is based on the definitions of satisfaction and the induction hypothesis. The most interesting cases are when the formula is a quantified. Suppose that $\varphi = \exists x \psi$.
  $\mathcal{M} \models \varphi$ iff $\mathcal{M}' \models \psi(c/x)$ for a $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ such that $\mathcal{M}' =_c \mathcal{M}$.
  By the induction hypothesis and previous lemma,
  $$\mathcal{M}' \models \psi(c/x) \text{ iff } \mathcal{M}^{'*} \models \psi(c/x) \text{ for a } \mathcal{M}^{'*} =_c \mathcal{M}^*.$$
  Therefore, $\mathcal{M}^* \models \exists x\, \psi$.
  In the case of $\varphi = \forall x\, \psi$ the proof is analogous.

Therefore, $\mathcal{M} \models_n \varphi$ iff $\mathcal{M}^* \models_m \varphi$. ∎

Having the notion of n-satisfiability, a new concept arises, that of logical n-consequence; also relative to the cardinality of the model in question.

Definition 2.12 (logical n-consequence)
Given a finite set of formulae Γ and a formula φ; we say that φ is a logical n-consequence of Γ iff for each structure $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ with $|\mathcal{D}| = n$, if $\mathcal{M} \models \Gamma$, then $\mathcal{M} \models \varphi$. In symbols, $\Gamma \models_n \varphi$.

The following result is the corresponding to Theorem 2.9 with respect to the notion of logical n-consequence. The proof our contribution.

Theorem 2.13
Let Γ be a set of formulae and φ a formula. $\Gamma \models_n \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is not n-satisfiable.

Proof. $\Longrightarrow$)(by reductio ad absurdum) Suppose that $\Gamma \models_n \varphi$ and that $\Gamma \cup \{\neg\varphi\}$ is n-satisfiable. So, there is a model $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ with $|\mathcal{D}'| = n$ such that:

$$\mathcal{M}' \models \Gamma \cup \{\neg\varphi\}$$

That is,

$$\mathcal{M}' \models_n \Gamma \text{ and } \mathcal{M}' \models_n \neg\varphi$$

But by hypothesis we have that any model of Γ with cardinality n is a model of φ, in particular $\mathcal{M}'$. Then,

$$\mathcal{M}' \models \varphi \text{ and } \mathcal{M}' \models \neg\varphi$$

But this is impossible. Then, $\Gamma \cup \{\neg\varphi\}$ is not n-satisfiable.
$\Longleftarrow$) Suppose that $\Gamma \cup \{\neg\varphi\}$ is not n-satisfiable; then, there is no model of $\Gamma \cup \{\neg\varphi\}$ with cardinality n. Also, let $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ be a structure such that $\mathcal{M} \models_n \Gamma$. But, given that φ is a formula, it verifies one of the following:

$$\mathcal{M} \models_n \varphi \text{ or } \mathcal{M} \models_n \neg\varphi$$

But by hypothesis we have that there is no model of cardinality n which satisfies Γ and $\neg\varphi$. So, $\mathcal{M} \models_n \varphi$.

■

Corollary 2.14
Let Γ be a set of formulae and φ a formula. If $\Gamma \models_n \varphi$, then $\Gamma \models_m \varphi$ for any m with $1 \leq m < n$.

Proof. Suppose that $\Gamma \not\models_m \varphi$; then $\Gamma \cup \{\neg\varphi\}$ is m-satisfiable and by theorem 2.9 we have that $\Gamma \cup \{\neg\varphi\}$ is n-satisfiable, contradicting $\Gamma \models_n \varphi$. So, $\Gamma \models_m \varphi$ for any m with $1 \leq m < n$.

■

3 N-Abduction Problem

We have now the necessary tools to tackle the problem of abduction in (decidable) first order formulae; we proceed by presenting abduction as a logical inference in $\mathcal{FOL}$ sets of formulae, those satisfiable in finite models.

Definition 3.1 (n-abductive problem)
Let Θ be a finite set of formulae of $\mathscr{L}$ minimally n-satisfiable ($1 \leq n < \omega$) and φ a formula of $\mathscr{L}$. We say that $\langle \Theta, \varphi \rangle$ is an n-abductive problem, which we denote as $\langle \Theta, \varphi \rangle_n$, if it verifies:

$$\Theta \not\models_n \varphi \text{ and } \Theta \not\models_n \neg\varphi$$

In a similar way, the solution of an n-abductive problem is given in terms of the concept of logical n-consequence. In the case of n-abductive problems, we are again interested in those solutions which verify the inferential requirement, which are also consistent with the theory and are explanatory.

Definition 3.2 (n-abductive solution)
Let $\langle \Theta, \varphi \rangle_n$ be an n-abductive problem and α a formula of $\mathscr{L}$; α is an n-abductive solution for $\langle \Theta, \varphi \rangle_n$ if it verifies:

1. $\Theta \cup \{\alpha\} \models_n \varphi$.
2. $\Theta \cup \{\alpha\}$ is minimally n-satisfiable ($\Theta \cup \{\alpha\} \not\models_n \perp$).
3. $\alpha \not\models_n \varphi$.

Observation 3.3
$\alpha \not\models_n \neg\varphi$ follows from 1 and 2 (analogous to observation 2.4).

4 N-tableaux

In this section we present an effective procedure that obtains solutions to n-abductive problems. This method is based on semantic tableaux, a logical framework introduced independently by Beth[3] and Hintikka[7]. We use the format and terminology of a later presentation, that found in [11]. One of its characteristics is that (non-literal) statements are classified into four classes, according to their syntax. This classification is adopted for N-tableaux.

Definition 4.1 (Stament Classification)
Let φ be a non-literal statement of $\mathscr{L}$.

1. φ is of *type* α if it is the conjunction of two $\mathscr{L}$ statements. That is, $\varphi = \psi \wedge \chi$. ψ and χ are the *α-subformulas* of φ.
2. φ is of *type* β if it is the conjunction of two $\mathscr{L}$ statemnets. That is, $\varphi = \psi \vee \chi$. ψ and χ are the *β-subformulas* of φ.
3. φ is of *type* γ if it is a quantified formula of a formula in $\mathscr{L}$, say $\varphi = \forall x \psi$, with ψ a formula of $\mathscr{L}$ whit x as the only free variable. Hence, the result after the substitution of the occurrences of variable x by a constant c in ψ is one statement. The set of these statements are denoted as *γ-subformulas* of φ.
4. φ is of *type* δ if it is the existential quantification of a formula of $\mathscr{L}$, say $\varphi = \exists x \psi$ such that ψ is a formula of $\mathscr{L}$ con $Vl(\psi) = \{x\}$. The set of statements which result from the substitution of x by a constant c, $\psi(c/x)$, are denoted *β-subformulas* of φ.

Before presenting the notion of N-tableau, we introduce some definitions and results which will be used in the proofs of N-tableaux properties.

Definition 4.2 (downward saturated set)
Let Γ be a finite set of formulae of first order logic and $\mathcal{D}_\Gamma$ the Herbrand universe of Γ. The set Γ is *downward saturated* if it verifies the following rules:

1. If Γ contains a formula α, than Γ contains its two α-subformulae.
2. If Γ contains a formula β, then Γ contains at least one of its β-subformulae.
3. If Γ contains a formula $\gamma = \forall x\ \psi$, than for each $c_i \in \mathcal{D}_\Gamma$, the γ-subformula $\psi(c_i/x)$ belongs to Γ.
4. If Γ contains a formula $\delta = \exists x\ \psi$, then Γ contains at least one δ-subformula $\psi(c/x)$ with $c \in \mathcal{D}_\Gamma$.

Definition 4.3 (Hintikka set)
An *(atomic) Hintikka set* is a downward saturated set such that no one formula and its negation belong to it.

Lemma 4.4
Let Γ be a Hintikka set; then Γ has a Herbrand model.

Proof. Let us define a Herbrand structure $\mathcal{H} = \langle \mathcal{D}_\Gamma, \mathcal{I}_\mathcal{H} \rangle$ for Γ according to the following rules:

1. $\mathcal{D}_\Gamma = \{c \mid$ being c a logical constant which appears in $\Gamma\}$
2. $\mathcal{I}_H(c_i) = c_i$ for each $c_i \in \mathcal{D}_\Gamma$.
3. $\mathcal{I}_H(P^k_m) = \{(c_1, \ldots, c_k) \in \mathcal{D}^k_\Gamma \mid P^n_m(c_1, \ldots, c_k) \in \Gamma\}$

Once defined $\mathcal{H} = \langle \mathcal{D}_\Gamma, \mathcal{I}_\mathcal{H} \rangle$, we only have to prove that $\mathcal{H} \models \varphi$ for every $\varphi \in \Gamma$. We proceed by induction on the complexity degree of φ in order to prove the following:
If $\varphi \in \Gamma$ then $\mathcal{H} \models \varphi$.

The base case is trivial. Let $\varphi = P^k_m(c_1, \ldots, c_k) \in \Gamma$; then from the definition of $\mathcal{H}$, it follows that $\mathcal{H} \models P^k_m(c_1, \ldots, c_k)$.

- If $\varphi = \alpha_1 \wedge \alpha_2$ (type α), by hypothesis we have that $\alpha_1, \alpha_2 \in \Gamma$ and given that the complexity degree of α_1 and α_2 is lesser than that of α, then $\mathcal{H} \models \alpha_1$ and $\mathcal{H} \models \alpha_1$. Hence, $\mathcal{H} \models \alpha_1 \wedge \alpha_2$.
- If $\varphi = \beta_1 \vee \beta_2$ (type β), then we have that $\beta_1 \in \Gamma$ or $\beta_2 \in \Gamma$. Without loss of generality, suppose that $\beta_1 \in \Gamma$; given that de complexity degree of β_1 is lesser than that of φ, by induction hypothesis we have that $\mathcal{H} \models \beta_1$. Hence, $\mathcal{H} \models \beta_1 \vee \beta_2$.
- If $\varphi = \forall x \psi$ (type γ) then $\psi(c_i/x) \in \Gamma$ is verified for every $c_i \in \mathcal{D}_\Gamma$. And as the complexity degree of the formulae $\psi(c_i/x)$ is lesser than that of φ, by induction hypothesis we have that $\mathcal{H} \models \psi(c_i/x)$ for every $c_i \in \mathcal{D}_\Gamma$. So, we conclude $\mathcal{H} \models \forall x \psi$.
- If $\varphi = \exists x \psi$ (type δ) then $\psi(c/x) \in \Gamma$ is verified for some $c \in \mathcal{D}_\Gamma$; given that the complexity degree of $\psi(c/x)$ is lesser than that of φ, by induction hypothesis we get $\mathcal{H} \models \psi(c/x)$. So, $\mathcal{H} \models \exists x \psi$.

We conclude that $\mathcal{H} \models \Gamma$. ∎

Definition 4.5 (N-tableau)
Let Γ be a finite set of formulae of $\mathscr{L}$ minimally n-satisfiable, with $1 \leq n < \omega$. A N-tableau of Γ, denoted by $\mathcal{T}_N(\Gamma)$, can be represented by means of a tree with the following properties:

a) Every node is labeled with a formula of $\mathscr{L}$.

b) We refer to a branch of a tableau as the set of formulae occurring on it, so we will denote them with uppercase Greek letters, in the same way we have done up to now for sets of formulae. The nodes of the initial branch are labeled with the formulae of $\Gamma \neq \varnothing$; we call this set the *initial set*.

c) The relations of right and left subtree are given by the following expansion rules: let Φ be a branch of the tableau of Γ and φ a formula on Ψ; we extend the branch Φ according to the following rules:

   1. If φ is of type α, the corresponding expansion rule is the α-*rule*, defined as follows:

$$\frac{\alpha_1 \wedge \alpha_2}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}}$$

   The application of this rule adds two new nodes to the branch Φ, labeled with the α-subformulae α_1 and α_2. In symbols: $\Phi + \alpha_1 + \alpha_2$.

   2. If φ is of type β, the corresponding rule is the β-*rule* defined as follows:

$$\frac{\beta_1 \vee \beta_2}{\beta_1 \mid \beta_2}$$

   The β-rule splits the current branch into two new branches, each of them labeled with one of the two β-subformulae β_1 and β_2. In symbols: $\Phi + \beta_1$ and $\Phi + \beta_2$.

3. If φ is of type γ, the *γ-rule* is applied, defined as follows:

$$\frac{\forall x\ \psi}{\begin{array}{c}\varphi(c_1/x)\\ \varphi(c_2/x)\\ \vdots\\ \varphi(c_n/x)\end{array}}$$

   This rule adds one node for any different constant which appear in the branch Φ, each one labeled with a γ-subformula $\varphi(c_i/x)$, being c_i a constant of Φ. That is, if $c_1, c_2, \ldots, c_n$ (with $n \geq 1$) are the constants which appear in the branch Φ (when there does not appear any constant, $n = 1$), then the result of applying the rule γ is the following: $\Phi + \varphi(c_1/x) + \varphi(c_2/x) + \ldots + \varphi(c_n/x)$.
   This rule has the peculiarity that remains active, that is to say, that whenever a new constant c appear in Φ as the result of the application of an expansion rule, we add a new node labeled with the corresponding γ-subformula to c, that is, $\varphi(c/x)$.
4. If φ is of type δ, the *δ_N-rule* is applied, defined as follows:

$$\frac{\exists x\varphi}{\varphi(c_1/x)|\ldots|\varphi(c_n/x)}$$

   where c_i $(1 \leq i \leq n)$ are the n constants of the language $\mathscr{L}$. The formulae $\varphi(c_i/x)$ $(1 \leq i \leq n)$ are called δ_N-subformulae of $\exists x\varphi$.

Observation 4.6
The rules α, β and γ are defined in the same way as in Beth's tableaux. The difference is in the rule corresponding to the formulae of type δ. It is trivial to verify that the new extension rule δ_N is sound, because by hypothesis we have that the set of formulae Γ is minimally n-satisfiable. Then, Γ has a canonical model $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ with cardinality n, so, $\mathcal{M} \models \varphi(c_i/x)$ for some $1 \leq i \leq n$, because $\mathcal{M}$ interprets n constants, each of which denotes one and only one of the objects in the universe of $\mathcal{D}$.

Definition 4.7 (closed N-tableau)
Let Γ be a finite set of formulae of $\mathscr{L}$ and $\mathcal{T}(\Gamma)$ a N-tableau of Γ. We say that a branch Φ of $\mathcal{T}(\Gamma)$ is *(atomically) closed* if an (atomic) formula φ and its negation appear in Φ. A N-tableau is *(atomically) closed* if all its branches are (atomically) closed.

Definition 4.8 (open N-tableau)
A branch Φ of a N-tableau is *open* if it is not closed. A N-tableau is *open* if it has at least one open branch.

In a similar way, we define a *saturated n-tableau* of a given set of formulae Γ as a tableau of Γ, in which it is not possible to make any extension.

Definition 4.9 (saturated N-tableau)
A branch Φ is saturated if the set containing all the formulae which appear in Φ is downward saturated. A N-tableau of a minimally n-satisfiable finite set Γ of formulae of $\mathscr{L}$ is saturated if all its branches (open or closed) are saturated.

Definition 4.10 (weight)
A branch of an N-tableau has *weight* n, for $n \geq 1$, if n is the number of (not repeated) constants occurring in the branch.

Observation 4.11
Any open and saturated branch Φ of a N-tableau of Γ, $\mathcal{T}_N(\Gamma)$, is a Hintikka set.

The interpretations taken into account in the construction of a N-tableau share the property that any element of the domain receives a unique name; there is one and only one constant to refer to each element of the domain. We refer to these interpretations as *canonical interpretations*.

Definition 4.12 (canonical interpretations)
Let $\mathscr{L}$ be a first order language and $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ an interpretation of $\mathscr{L}$. $\mathcal{M}$ is an canonical interpretation if for any $c, c' \in Cons(\mathscr{L})$ and any $a \in \mathcal{D}$, if $\mathcal{I}(c) = \mathcal{I}(c') = a$ then $c = c'$. That is to say, if the interpretation function $\mathcal{I}$ restricted to the constants is injective.

Note that given a set of formulae Γ of $\mathscr{L}$ minimally n-satisfiable with a model $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ it is always possible to make a transformation such that we get a set of formulae minimally n-satisfiable Γ^* of a language $\mathscr{L}^*$ and a canonical $\mathscr{L}^*$-model $\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$, such that $\mathcal{M} \models \Gamma$ iff $\mathcal{M}^* \models \Gamma^*$. We call to this transformation n-standarization.

Definition 4.13 (n-standarization)
Let Γ be a set of formulae of $\mathscr{L}$ with m constants $c_1, c_2, \ldots, c_m$ and $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ a $\mathscr{L}$-model of Γ, such that $|D| = n$, $D = \{a_1, \ldots, a_n\}$. An n-standarization of $\langle \mathscr{L}, \Gamma, \mathcal{M} \rangle$ consists of a language $\mathscr{L}^*$, a set of formulae Γ^* of the language $\mathscr{L}^*$ and a $\mathscr{L}^*$-model $\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$ such that:

- $\mathscr{L}^*$ has n constants $d_1, \ldots, d_n$ where each $d_j \in N_j$ with $N_j = \{c_i \mid \mathcal{I}(c_i) = a_j\}$. If $N_j = \varnothing$, d_j is a new constant.
- $\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$, with
$$\begin{array}{lcl} D^* & = & D \\ \mathcal{I}^* \restriction_{Pred(\mathscr{L})} & = & \mathcal{I} \\ \mathcal{I}^*(d_j) & = & a_j \end{array}$$
- $\Gamma^* = \{\varphi^* \mid \varphi \in \Gamma\}$, where φ^* is recursively defined as follows:
  – $P(t_1, \ldots, t_n)^* = P(t_1^*, \ldots, t_n^*)$.
  – $(\neg\varphi)^* = \neg\varphi^*$
  – $(\varphi \wedge \psi)^* = \varphi^* \wedge \psi^*$
  – $(\varphi \vee \psi)^* = \varphi^* \vee \psi^*$
  – $(\forall x \varphi)^* = \forall x \varphi^*$
  – $(\exists x \varphi)^* = \exists x \varphi^*$
  For terms we have the following:
  – $x^* = x$ being x a variable.
  – $c^* = d_j$ si $\mathcal{I}(c) = a_j$.

Theorem 4.14
Let φ be a formula of $\mathscr{L}$, $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ a $\mathscr{L}$-interpretation and $\langle \mathscr{L}^*, \varphi^*, \mathcal{M}^* \rangle$ a n-standarization of $\langle \mathscr{L}, \varphi, \mathcal{M} \rangle$. Then, $\mathcal{M} \models \varphi$ iff $\mathcal{M}^* \models \varphi^*$.

Proof. Induction on φ. ■

The above theorem suggests that all models are canonical. In fact, when considering a model of size n we can suppose that the underlying language has n constants, each of which denotes an element of the domain. In what follows, we take that all models are canonical ones, and that the number of constants of the language is equal to the cardinality of the domain.

4.1 N-tableaux Soundness

N-tableaux can be used to find models for sets of formulae minimally n-satisfiable, provided the value of n is given, for it is necessary to apply the extension δ_N-rule.

Lemma 4.15
Let Γ be a finite set of formulae of $\mathscr{L}$ minimally n-satisfiable. If $\mathcal{T}_N(\Gamma)$ is a saturated N-tableau of Γ with a saturated and open branch Φ of weight n, then it is possible to construct (out of Φ) a model of Γ with cardinality n.

Proof. Let Φ be an open and saturated branch of $\mathcal{T}_N(\Gamma)$ with weight n. Φ is a Hintikka set. By Lemma 4.4, Φ has a Herbrand model $\mathcal{H} = \langle \mathcal{D}_\Phi, \mathcal{I}_\mathcal{H} \rangle$ with cardinality n, that is, $\mathcal{H} \models_n \varphi_i$ for every formula φ_i of Φ, in particular for those of Γ. So, $\mathcal{H}$ is a model of Γ. ∎

Lemma 4.16
Let Γ be a finite set of formulae of $\mathscr{L}$. If Γ is minimally n-satisfiable, then every saturated $\mathcal{T}_N(\Gamma)$ verifies the following conditions:

1. All the branches of weight m, with $1 \leq m < n$, are closed.
2. $\mathcal{T}_N(\Gamma)$ has an open branch of weight n.

Proof. Let Γ be a finite set of formulae minimally n-satisfiable. Let $\mathcal{T}_N(\Gamma)$ be a saturated N-tableau. Suppose that $\mathcal{T}_N(\Gamma)$ contains an open and saturated branch of weight m ($1 \leq m < n$). By Lemma 4.15, it is possible to construct a model of cardinality m, contradicting Γ is minimally n-satisfiable. So, the first condition is verified, all the branches of weight lesser than n are closed.

Now, suppose that $\mathcal{T}_N(\Gamma)$ does not contain any open branch of weight n. By hypothesis we know that there is a canonical structure $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ such that $|\mathcal{D}| = n$ and $\mathcal{M} \models_n \Gamma$, because the result of applying any of the extension rules of the N-tableaux to a satisfiable set of formulae causes that at least one of the branches remains open[3], by adding new formulae satisfied by $\mathcal{M}$. Hence, it cannot occur that all the branches are closed,[4] so at least one of the branches of weight n should be open. ∎

Theorem 4.17
A finite set of formulae Γ is minimally n-satisfiable iff there is a Hintikka set $\mathcal{T}_N(\Gamma)$ with an open and saturated branch of weight n, and all branches of weight lesser than n are closed.

Proof. The proof is direct by Lemma 4.16 and observation 4.11. ∎

5 N-Abductive Reasoning via N-tableaux

In this section we characterize N-tableaux as an effective procedure to search for solutions to n-abductive problems.

Theorem 5.1
If $\langle \Theta, \varphi \rangle_n$ is an n-abductive problem with $1 \leq n < \omega$, then there is a saturated N-tableau, $\mathcal{T}_N(\Theta \cup \{\neg\varphi\})$ which contains an open branch of weight n.

[3] This is a consequence of the soundness of the extension rules.

[4] Note that no one branch has weight greater than n, because the maximum number of introduced constants is fixed (n).

PROOF. By definition of $\langle\Theta,\varphi\rangle_n$, we have that $\Theta \not\models_n \varphi$ and that Θ is a minimally n-satisfiable. Therefore, by the refutation principle we have that $\Theta \cup \{\neg\varphi\}$ is a minimally n-satisfiable set, so that by Theorem 4.17, each saturated N-tableau, $(\mathcal{T}_N(\Theta \cup \{\neg\varphi\})$, has an open branch of weight n. ∎

THEOREM 5.2
Let $\langle\Theta,\varphi\rangle_n$ be an n-abductive problem and α a formula of $\mathscr{L}$; α is an n-abductive solution of $\langle\Theta,\varphi\rangle_n$ iff:

1. Any saturated N-tableau of $\Theta \cup \{\neg\varphi\} \cup \{\alpha\}$ is closed, that is, all its branches are closed.
2. Any saturated N-tableau of $\Theta \cup \{\alpha\}$ has at least one open branch of weight n and every branch of weight m, $1 \leq m < n$, is closed.
3. Any saturated N-tableau of $\{\alpha\} \cup \{\neg\varphi\}$ has at least one open branch of weight m, with $1 \leq m \leq n$, and all the branches with a lesser weight are closed.

PROOF. 1. By the definition of n-abductive solution we have that $\Theta \cup \{\alpha\} \models_n \varphi$, but this is verified iff the set $\Theta \cup \{\alpha\} \cup \{\neg\varphi\}$ is not minimally n-satisfiable. By Theorem 4.17, this is satisfied iff any saturated $\mathcal{T}_N(\Theta \cup \{\neg\varphi\} \cup \{\alpha\})$ is closed.

2. By definition, $\Theta \cup \{\alpha\}$ is minimally n-satisfiable; again this happens iff any saturated $\mathcal{T}_N(\Theta \cup \{\alpha\})$ contains an open branch of weight n, and all the branches with a lesser weight are closed (theorem 4.17).
3. By definition of n-abductive solution we have that $\alpha \not\models_n \varphi$; this happens iff $\{\alpha\} \cup \{\neg\varphi\}$ is n-satisfiable. Let m be the value, $1 \leq m \leq n$, such that $\{\alpha\} \cup \{\neg\varphi\}$ is minimally m-satisfiable (Theorem 2.9 guarantees that such m exists). Hence, every saturated N-tableau of $\{\alpha\} \cup \{\neg\varphi\}$ contains an open branch of weight m.

   Now, let $\mathcal{T}_N(\{\alpha\} \cup \{\neg\varphi\})$ be a saturated N-tableau with an open branch of weight m, being m minimal; then $\{\alpha\} \cup \{\neg\varphi\}$ is a minimally m-satisfiable set, by Theorem 4.17. Given that n is greater than m, then $\{\alpha\} \cup \{\neg\varphi\}$ is a n-satisfiable set, and then $\alpha \not\models_n \varphi$. ∎

N-tableaux seem to be the first example of an effective procedure to solve an n-abductive problem $\langle\Theta,\varphi\rangle_n$. The process involves first the construction of a saturated N-tableau of $\Theta \cup \{\varphi\}$, and then it proceeds to close it, for what the closing set of any of its open branches is necessary to be obtained.

DEFINITION 5.3 (closing set)
Let Φ be a branch of a N-tableau and $Lit(\Phi)$ the set of literals which appear in Φ. The closing set of Φ, denoted by $CC(\Phi)$, is the set which contains the complementary literals of all the literals of $Lit(\Phi)$.

Once the closing set has been formally defined, the following is a process to find solutions for an n-abductive problem $\langle\Theta,\varphi\rangle_n$, given a saturated N-tableau of $\Theta \cup \{\neg\varphi\}$ which the following open branches $\Phi_1, \ldots, \Phi_k$.

1. For each branch Φ_i, $1 \leq i \leq k$, the set $CC(\Phi_i)$ is obtained.
2. If the intersection of all the $CC(\Phi_k)$, $1 \leq i \leq k$, is non-empty, then it should be verified whether any of its literal elements is an n-abductive solution. Else, this process ends, as there are no literal-solutions (go to step 3)).

3. If there are not literal solutions, it should be verified whether there are conjunctive solutions, those which are formed by the conjunction of literals of the closing sets. It should be verified whether each of the proposed conjunctions is actually an n-abductive solution.

Observation 5.4
If an abductive solution has the following form:

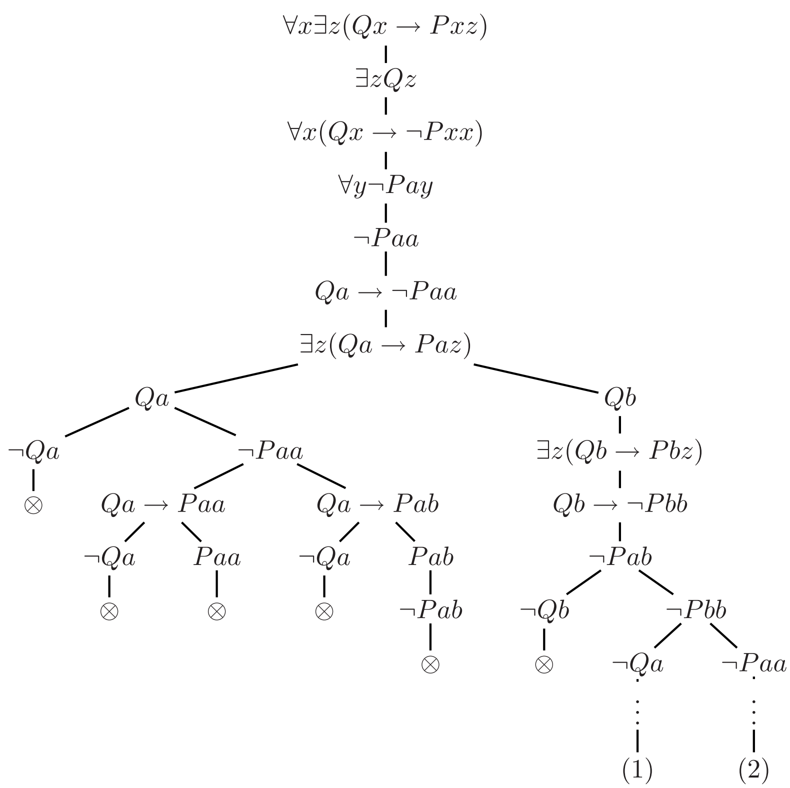
$$\varphi(c_1/x) \wedge \varphi(c_2/x) \wedge \ldots \wedge \varphi(c_n/x)$$

then the corresponding n-abductive solution is equivalent to the following quantified formula: $\forall x \varphi$.

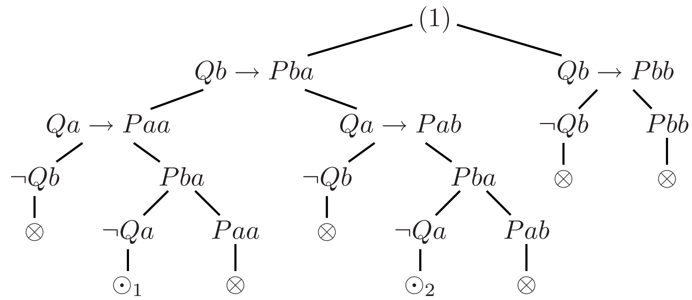
In what follows, we present a couple of examples, illustrative of the n-abductive effective procedure we propose in this paper.

Example 5.5
Let $\Theta = \{\forall x \exists z(Qx \to Pxz),\ \exists z Qz,\ \forall x(Qx \to \neg Pxx)\}$ be a theory minimally 2-satisfiable and $\varphi = \exists y Pay$ a fact. Find a solution for the 2-abductive problem $\langle \Theta, \varphi \rangle_2$.

Solution. Let us construct the N-tableau corresponding to $\Theta \cup \{\neg\varphi\}$.



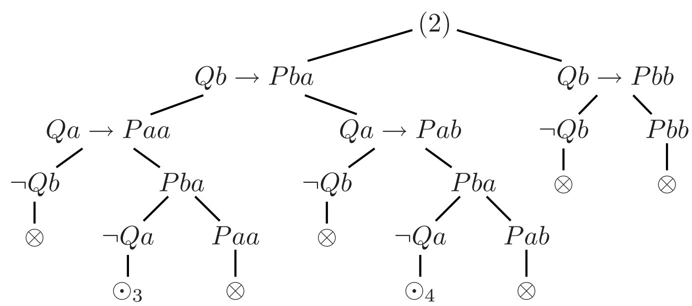
Before we proceed with the construction of the N-tableau, note that all branches of weight 1 were closed by applying only extension rules to formulae of Θ. We extend the tree from left to right. Also, we use the symbols $\otimes$ and $\odot$ to represent that a branch is closed or saturated and open, respectively.



¿From the extension of branch (1), we get two open and saturated branches ($\odot_1$ y $\odot_2$). The set of literals of any of them is the same, namely:

$$\{\neg Paa, Qb, \neg Pab, \neg Pbb, \neg Qa, Pba\}$$

The result of extending the branch (2) of the initial N-tableau is the following:



Again, we obtain two open and saturated branches ($\odot_3$ and $\odot_4$), with the same set of literals as those obtained from (1). So, the closing set of any of them is the same:

$$\{Paa, \neg Qb, Pab, Pbb, Qa, \neg Pba\}$$

We only need to verify which of them is a solution to the problem $\langle \Theta, \varphi \rangle_n$. Neither Pab nor Paa can be explanations, because they are trivial. The literals $\neg Qb$, Qa, Pbb and $\neg Pba$ are solutions to $\langle \Theta, \varphi \rangle_n$.

Example 5.6
Given the theory $\Theta = \{\forall x \exists y (Px \rightarrow Qy \wedge Rxy),\ \exists x(Qx \wedge \neg Px),\ \exists x(Px \wedge Qx),\ \exists x(Px \wedge \neg Qx),\ \forall x \neg Rxx\}$, which is minimally 3-satisfiable, and the fact $\varphi = Qc \wedge Pc$, find a solution for the 3-abductive problem $\langle \Theta, \varphi \rangle_3$.

Solution. The initial part of the saturated N-tableau for $\Theta \cup \{\neg\varphi\}$ is the following:

$$
\begin{array}{c}
\forall x\exists y(Px \to Qy \land Rxy)\\
|\\
\exists x(Px \land \neg Qx)\\
|\\
\exists x(Px \land Qx)\\
|\\
\exists x(Qx \land \neg Px)\\
|\\
\forall x\neg Rxx\\
|\\
\neg(Qc \land Pc)\\
|\\
\exists y(Pc \to Qy \land Rcy)\\
|\\
\neg Rcc\\
\diagup \qquad \diagdown\\
\begin{array}{cc}
\neg Qc & \neg Pc\\
\vdots & \vdots\\
| & |\\
(1) & (2)
\end{array}
\end{array}
$$

By expanding the N-tableau until it is saturated, we get the following branches:

$$
\begin{array}{rcl}
Lit(\odot_{(1.1.1.1)}) & = & \{\neg Qc,\ Pa,\ Qa,\ \neg Pb,\ Qb,\ Pc,\ Rca,\ Rab\}\\
Lit(\odot_{(1.1.1.2)}) & = & \{\neg Qc,\ Pa,\ Qa,\ \neg Pb,\ Qb,\ Pc,\ Rcb,\ Rab\}
\end{array}
$$

$$
\begin{array}{rcl}
Lit(\odot_{(1.1.2.1)}) & = & \{\neg Qc\ Pa,\ Qa,\ \neg Pb,\ Qb,\ Pc,\ Rba,\ Rca,\ Rab\}\\
Lit(\odot_{(1.1.2.2)}) & = & \{\neg Qc,\ Pa,\ Qa,\ \neg Pb,\ Qb,\ Pc,\ Rba,\ Rcb,\ Rab\}
\end{array}
$$

$$
\begin{array}{rcl}
Lit(\odot_{(1.2.1.1)}) & = & \{\neg Qc,\ Pb,\ Qb,\ \neg Pa,\ Qa,\ Pc,\ Rca\}\\
Lit(\odot_{(1.2.1.2)}) & = & \{\neg Qc,\ Pb,\ Qb,\ \neg Pa,\ Qa,\ Pc,\ Rcb\}\\
Lit(\odot_{(1.2.2.1)}) & = & \{\neg Qc,\ Pb,\ Qb,\ \neg Pa,\ Qa,\ Pc,\ Rab,\ Rca\}\\
Lit(\odot_{(1.2.2.2)}) & = & \{\neg Qc,\ Pb,\ Qb,\ \neg Pa,\ Qa,\ Pc,\ Rab,\ Rcb\}
\end{array}
$$

$$
\begin{array}{rcl}
Lit(\odot_{(2.1.1)}) & = & \{\neg Pc,\ Pa,\ Qa,\ Pb,\ \neg Qb,\ Qc,\ \ Rac\}\\
Lit(\odot_{(2.1.2)}) & = & \{\neg Pc,\ Pa,\ Qa,\ Pb,\ \neg Qb,\ Qc,\ \ Rac,\ Rca\}
\end{array}
$$

$$
\begin{array}{rcl}
Lit(\odot_{(2.2.1.1)}) & = & \{\neg Pc,\ Pb,\ Qb,\ Pa,\ \neg Qa,\ Qc,\ Rbc,\ Rab,\}\\
Lit(\odot_{(2.2.1.2)}) & = & \{\neg Pc,\ Pb,\ Qb,\ Pa,\ \neg Qa,\ Qc,\ Rbc,\ Rab,\ Rcb\}\\
Lit(\odot_{(2.2.2.1)}) & = & \{\neg Pc,\ Pb,\ Qb,\ Pa,\ \neg Qa,\ Qc,\ Rbc,\ Rac,\ \}\\
Lit(\odot_{(2.2.2.2)}) & = & \{\neg Pc,\ Pb,\ Qb,\ Pa,\ \neg Qa,\ Qc,\ Rbc,\ Rcb,\ \}
\end{array}
$$

Once we have a saturated N-tableau for the 3-abductive $\langle\Theta, \varphi\rangle_3$, we only have to find a solution from the closing sets os the open and saturated branches. We list them:

$$
\begin{array}{lcl}
CC(\odot_{(1.1.1.1)}) & = & \{Qc,\ \neg Pa,\ \neg Qa,\ Pb,\ \neg Qb,\ \neg Pc,\ \neg Rca,\ \neg Rab\} \\
CC(\odot_{(1.1.1.2)}) & = & \{Qc,\ \neg Pa,\ \neg Qa,\ Pb,\ \neg Qb,\ \neg Pc,\ \neg Rcb,\ \neg Rab\} \\
CC(\odot_{(1.1.2.1)}) & = & \{Qc\ \neg Pa,\ \neg Qa,\ Pb,\ \neg Qb,\ \neg Pc,\ \neg Rba,\ \neg Rca,\ \neg Rab\} \\
CC(\odot_{(1.1.2.2)}) & = & \{Qc,\ \neg Pa,\ \neg Qa,\ Pb,\ \neg Qb,\ \neg Pc,\ \neg Rba,\ \neg Rcb,\ \neg Rab\} \\
CC(\odot_{(1.2.1.1)}) & = & \{Qc,\ \neg Pb,\ \neg Qb,\ Pa,\ \neg Qa,\ \neg Pc,\ \neg Rca\} \\
CC(\odot_{(1.2.1.2)}) & = & \{Qc,\ \neg Pb,\ \neg Qb,\ Pa,\ \neg Qa,\ \neg Pc,\ \neg Rcb\} \\
CC(\odot_{(1.2.2.1)}) & = & \{Qc,\ \neg Pb,\ \neg Qb,\ Pa,\ \neg Qa,\ \neg Pc,\ \neg Rab,\ \neg Rca\} \\
CC(\odot_{(1.2.2.2)}) & = & \{Qc,\ \neg Pb,\ \neg Qb,\ Pa,\ \neg Qa,\ \neg Pc,\ \neg Rab,\ \neg Rcb\} \\
CC(\odot_{(2.1.1)}) & = & \{Pc,\ \neg Pa,\ \neg Qa,\ \neg Pb,\ Qb,\ \neg Qc,\ \ Rac\} \\
CC(\odot_{(2.1.2)}) & = & \{Pc,\ \neg Pa,\ \neg Qa,\ \neg Pb,\ Qb,\ \neg Qc,\ \neg Rac\ \neg Rca\} \\
CC(\odot_{(2.2.1.1)}) & = & \{Pc,\ \neg Pb,\ \neg Qb,\ \neg Pa,\ Qa,\ \neg Qc,\ \neg Rbc,\ \neg Rab\} \\
CC(\odot_{(2.2.1.2)}) & = & \{Pc,\ \neg Pb,\ \neg Qb,\ \neg Pa,\ Qa,\ \neg Qc,\ \neg Rbc,\ \neg Rab,\ \neg Rcb\} \\
CC(\odot_{(2.2.2.1)}) & = & \{Pc,\ \neg Pb,\ \neg Qb,\ \neg Pa,\ Qa,\ \neg Qc,\ \neg Rbc,\ \neg Rac\ \} \\
CC(\odot_{(2.2.2.2)}) & = & \{Pc,\ \neg Pb,\ \neg Qb,\ \neg Pa,\ Qa,\ \neg Qc,\ \neg Rbc,\ \neg Rcb\ \}
\end{array}
$$

The intersection of these sets is empty. So, there are not literals solutions. We need to construct conjunctive solutions.

By using the proposed procedure, it is possible to construct the following explanations:

$$
\begin{array}{c}
\neg Qb \wedge \neg Pa \\
\neg Qa \wedge \neg Pb
\end{array}
$$

We could find more solutions, but these have the property of being minimal.

6 Conclusions

Abduction in first order logic may be regarded as an intractable problem. We can see that from the very definition of what an abductive problem $\langle \Theta, \varphi \rangle$ is, for it requires that $\Theta \cup \{\neg\varphi\}$ is satisfiable, or the equivalent, that $\Theta \not\models \varphi$, and this is undecidable in $\mathcal{FOL}$. But the undecidability of first order logic has not been an obstacle to study the problem of logical consequence from an algorithmic point of view, using some fragments of $\mathcal{FOL}$ for which the problem of logical consequence is decidable (c.f. [5]). On the other hand, the advances in speed and computational capabilities of modern computers lead to a reinforcement of automatic reasoning by means of the implementation of many provers based on first order logic as well as on higher order logics (Otter and Isabelle, as a couple of examples).

Based on these ideas, and following [10, 2], we tackle the problem of first order abduction in a way that makes sense in those decidable fragments of $\mathcal{FOL}$ with the finite model property. To carry out our proposal, we propose a refinement of the traditional notions of satisfaction and logical consequence, which we call n-satisfaction and logical n-consequence, as they are relative to domains of n cardinality.

We perform abductive reasoning for sets of decidable formulae of $\mathcal{FOL}$ with finite models, by means of the notion of n-satisfiability. We define what is an n-abductive problem by means of logical n-consequence. A first version of this is found in [10, 2], where the tableaux independently proposed by [4] and[6] are used as a mechanical procedure to obtain explanations for n-abductive problems in the case that the abductive problem $\langle \Theta, \varphi \rangle_n$ satisfies the following conditions: both φ and α (the explanation) should be literals. In contrast, our approach does not require such restrictions; and our notion of n-consequence in defined a more detailed way.

We propose a modification of the method of Beth's semantic tableaux in order to obtain an effective procedure to find solutions for n-abductive problems. In place of the definition of the extension δ-rule of Beth's tableaux, we offer a new extension rule for formulae of type δ, which we call δ_N-rule. This rule depends on the cardinality of the domain too. We call N-tableaux the tableaux constructed by means of this new extension rule and the usual α, β and γ rules. We prove soundness and completeness of N-tableaux, by means of the notion of n-satisfiability.

Finally, we mention some suggestions for further research:

- Study properties (such as structural rules) of the notions of n-satisfiability and logical n-consequence.
- Develop heuristics which allow to find solutions for n-abductive problems through N-tableaux or DB-tableaux, in a way that it is possible to implement the results presented in this paper within a reasonable effort (non exponential) of space and time.
- Explore those decidable fragments of $\mathcal{FOL}$ with the finite model property, in order to find algorithms or heuristics based on the forms of the formulae which constitute an n-abductive problem.

Acknowledgments

The development of this paper involved several meetings in Mexico and Spain. The first author was supported by a National Council of Science and Technology (CONACYT) grant (number 172909) for master studies at the National Autonomous University of Mexico (UNAM).

References

[1] Aliseda Llera, A. Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence. Institute for Logic, Language and Computation, Amsterdam. 1997.

[2] Aliseda, A. *Abductive Reasoning: Logical Investigations into Discovery and Explanation*, volume 330, *Synthese Library*. Springer, 2005.

[3] Beth E.W. Semantic Entailment and Formal Derivability. *Mededelingen der Kon. Ned. Akad. v. Wet.* Vol 18, no.13. pp. 309-342. 1955

[4] Boolos, G. 1984., Trees and finite satisfiability: Proof of a conjecture of Burgess. *Notre Dame Journal of Formal Logic* Vol. 25 pp. 193-197. 1984.

[5] Börger, E., Grädel, E., Gurevich, Y. The Classical Decision Problem. Second printing. Springer Verlag, 2001.

[6] Díaz E. 1993. Árboles Semánticos y Modelos Mínimos". In E.Pérez (editor): *Actas del I Congreso de la Sociedad de Lógica, Metodología y Filosofía de la Ciencia en España*. Universidad Complutense de Madrid, pp. 40-43. 1993.

[7] Hintikka J., A New Approach to Sentencial Logics. *Societas Scientiarum Fennica Commentationes Physico-Mathematicae*. Vol. 17, No. 3. 1953.

[8] Mayer M.C. and Pirri F., First Order Abducction via Tableaux and Sequent Calculi. *Bulletin of The I.G.P.L.*, Vol. 1. pp.99-117. 1993.

[9] Nepomuceno A., Tablas semánticas y Metalógica (el caso de la lógica de segundo orden). *Crítica*. Vol. XXXXI, No. 93, pp.21-47.

[10] Nepomuceno A., Scientific Explanation and Modified Semantic Tableaux. En Magnani. L., Nerssessian, N.J., Pizzi, C. (Eds). Logical and Computational Aspects of Model-Based Reasoning. Applied Logic Series. pp.181-198. Kluwer 2002.

[11] R.M. Smullyan. *First Order Logic*. Springer Verlag, 1968.

[12] Stickel M., Upside-Down Meta-Interpretation of the Model Elimination Theorem-Proving Procedure for Deduction and Abduction. Technical Report. Institute for New Generation Computer Technology. (ICOT) TR-664. 1991.

Received December 31, 2005