First Order Abduction Via Tableau and Sequent Calculi

Article · January 1997

DOI: 10.1093/jigpal/1.1.99 · Source: CiteSeer

Marta Cialdea Mayer
Università Degli Studi Roma Tre

Fiora Pirri
Sapienza University of Rome

First order abduction via tableau and sequent calculi

Marta Cialdea Mayer[1,2] Fiora Pirri[1]

[1]Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza"
via Salaria 113, 00198 Roma, Italia
[2]Istituto di Informatica, Università degli Studi di Ancona
via delle Brecce Bianche, 60131 Ancona, Italia

e-mail: {cialdea,pirri}@assi.ing.uniroma1.it

Abstract

The formalization of abductive reasoning is still an open question: there is no general agreement on the boundary of some basic concepts, such as preference criteria for explanations, and the extension to first order logic has not been settled.

Investigating the nature of abduction outside the context of resolution based logic programming still deserves attention, in order to characterize abductive explanations without tailoring them to any fixed method of computation. In fact, resolution is surely not the best tool for facing meta-logical and proof-theoretical questions.

In this work the analysis of the concepts involved in abductive reasoning is based on analytical proof systems, i.e. tableaux and Gentzen-type systems. A proof theoretical abduction method for first order classical logic is defined, based on the sequent calculus and a dual one, based on semantic tableaux. The methods are sound and complete and work for full first order logic, without requiring any preliminary reduction of formulae into normal forms. In the propositional case, two different characterizations are given for abductive explanations, each of them being the declarative counterpart of a different algorithm for the generation of explanations. The first one corresponds to the generation of *the whole set* of minimal and consistent explanations, where minimality is checked by comparison with the other elements of the set. The second characterization corresponds to a (non-deterministic) algorithm for generating *a single* minimal explanation that is consistent with the theory.

The first order versions of the abductive systems make use of unification and dynamic herbrandization/skolemization of formulae. The construction of the abduced formula is pursued by means of de-skolemization. The first order methods are very loose in discarding explanations that are not minimal. In fact, the question of minimality in first order abduction is a main issue. As usually defined, minimality is undecidable, for two different reasons: (i) determining whether an explanation is better than another one is in general undecidable; (ii) the set of explanations may be infinite. Moreover, because of (ii), a minimal element may not exist. Such problems suggest that the minimality requirement should be relaxed, possibly defining it w.r.t. a stronger relation than logical consequence.

1 Abductive reasoning

1.1 Preliminaries

Abduction is a form of reasoning that infers premises from a conclusion. Its characteristic logical schema is the inference of φ from $\varphi \rightarrow \psi$ and ψ. It is an unsound form of inference, that reflects some forms of commonsense reasoning [14, 15, 18], where causes for events are to be hypothesized, and diagnostic reasoning [16]. More generally, abductive reasoning is a way to solve problems where an observed event φ is not explained by the presently adopted theory Θ and an explanation for φ has to be looked for.

Precisely, an *abduction problem* is given by a background theory Θ (a set of formulae) and a formula φ such that:
1) $\Theta \not\models \varphi$
2) $\Theta \not\models \neg\varphi$
A solution of the problem given by the pair $\langle \Theta, \varphi \rangle$ is to be looked for among the formulae α such that $\Theta \cup \{\alpha\} \models \varphi$.

In abductive reasoning explanations are required to respect some fundamental conditions, in order to be accepted as "interesting". Although there is no general agreement on the exact boundary between interesting and non-interesting explanations, the following three reasonable restrictions are usually imposed on explanations α for an abduction problem $\langle \Theta, \varphi \rangle$:

(i) α is consistent with Θ (or Θ-*consistent*), i.e. $\Theta \not\models \neg\alpha$.

(ii) α is a *minimal* explanation for the abduction problem $\langle \Theta, \varphi \rangle$, i.e. for any formula β, if $\Theta \cup \{\beta\} \models \varphi$ and $\alpha \models \beta$, then $\models \beta \equiv \alpha$.

(iii) α has some restricted syntactical form; for example, it is a prenex formula whose matrix is a conjunction of literals.

The syntactical restriction imposed on explanations in this work is stated in the following definitions.

Definition 1 (C-formulae) *A formula α is a (first-order) C-formula if it is built up from literals using only quantifiers and conjunction.*

Definition 2 (Explanations for abduction problems) *α is an explanation for the abduction problem $\langle \Theta, \varphi \rangle$ if it is a C-formula in the language of $\Theta \cup \{\varphi\}$ and $\Theta \cup \{\alpha\} \models \varphi$.*

The reason why explanations are not required to be prenex is that this fact simplifies both definitions and proofs. However, equivalent explanations are considered as identical, so that explanations are in fact equivalence classes of formulae. When an explanation is required to be either minimal or consistent with the theory, it will be explicitly stated.

In what follows, whenever we speak of minimality in a set of formulae, it is intended minimality with respect to $\models$.

1.2 Proof theoretical methods for performing abduction

The development of modern logic highlighted the fundamental duality between abduction and deduction, due to the fact that, if Θ is a logical theory and φ an observed fact, then for any α (in any logic where sentences can "cross" the logical entailment symbol $\models$):

$$\Theta, \alpha \models \varphi \quad iff \quad \Theta, \neg\varphi \models \neg\alpha$$

The difference lies in that, while deductive problems usually consist in verifying whether a given formula follows from a theory (possibly instantiating some variables), abduction problems are generative. Thus, as pointed out since the earliest papers on modern abduction [13, 17, 5], any deductive system that can be used not only to test, but also to generate consequences can be used to perform abduction.

Most proof theoretical methods for performing abduction are based on resolution (see for example [12, 13, 5, 9]). As a prerequisite for the use of resolution based methods, the theory and the negation of the observation must first of all be transformed into clause form. Moreover, works on abduction in the context of logic programming are often influenced by the linear resolution view, even in the definition of the basic concepts involved.

The above sketchy observations suggest that investigating the nature of abduction outside the context of resolution and logic programming still deserves attention, in order to characterize abductive explanations without tailoring them to any fixed method of computation. Moreover, the fact that many of the questions that should be addressed are meta-logical and proof-theoretical in nature suggests the use of proof systems that are, in that respect, more suited than resolution.

This work proposes to base the analysis of the concepts invoved in abductive reasoning on non-resolution logical systems, i.e. tableaux and Gentzen systems, whose importance in automated reasoning has been often neglected in the computer science community (the relation between these methods and resolution has been clearly analysed in [1]). Classical tableau and sequent calculi enjoy of *analicity*, a feature that resolution lacks and that makes proof theoretical investigations clearer. They are especially promising tools to deal with abduction: the interpretation represented by a given branch of a tableau or by a sequent is clearly a partial interpretation, and abduction can be framed in the context of three valued semantics [3].

In the rest of the work a proof theoretical abduction method for first order classical logic is defined, based on the sequent calculus and a dual one, based on semantic tableaux. The methods work for full first order logic, without requiring any preliminary reduction of formulae into normal forms. Soundness and completeness are established. In the propositional case, where the generation of the set of explanations is bound to terminate, two different characterizations are given for abductive explanations. Both identify explanations on the basis of a given set of tableaux branches (leaves of a sequent derivation tree) and each of them is the declarative counterpart of a different algorithm for the generation of explanations. The first one corresponds to the generation of *the whole set* of minimal and Θ-consistent explanations, built by an incremental method that uses the branches (leaves) one by one and discards them as they are used. Minimal explanations are singled out by comparision with the other elements of the whole set. The second characterization corresponds to a (non-deterministic) algorithm for generating *a single* minimal explanation that is consistent with the theory. Such a method requires that a given set of tableaux branches (leaves of a sequent derivation tree) is stored and used till the algorithm terminates.

The first order versions of the abductive systems make use of unification and dynamic herbrandization/skolemization of formulae. The construction of the abduced formula is pursued by means of de-skolemization. The undecidability of first order logic reflects on the fact that it may be impossible to terminate the construction of a tableau (derivation tree) and, therefore, the process of generation of explanations may not terminate. Consequently, the set of explanations may be infinite. Therefore, it is obvious that determining whether a given explanation is minimal is in general undecidable (even w.r.t. subsumption), and also that a minimal element may not exist. The method proposed here constructs the set of explanations in an incremental manner and the minimality check is very loose. In fact, the problem of identifying a reasonable relaxation of minimality w.r.t.$\models$, that ensures decidability, deserves a different work.

2 Abduction via sequent calculi and tableaux: the propositional case

In this section we are going to define a sound and complete propositional method for computing minimal and Θ-consistent explanations for an abduction problem. We consider a propositional language $\mathcal{L}_0$, containing two distinct propositional letters, *true* and $false$. Formulae, clauses and literals are defined as usual. Clauses and conjunctions of literals will sometimes be identified with the set of their literals, so that set operations on clauses or conjunctions of literals are allowed. We make the convention that the empty disjunction is equivalent to the atom $false$ and the empty conjunction to the atom *true*. Two disjunctions (conjunctions) are considered equal - and the equality sign '=' will be used - if they disjoin (conjoin) sets of equal elements.

2.1 Propositional semantic tableaux and sequent calculus

The propositional tableaux method [6] and Gentzen deduction system for classical logic [8] are essentially the same calculus considered from two different standpoints. We shall define propositional semantic tableaux as introduced by

Fitting [6], while the version of signed tableaux will be considered in order to highlight the substantial identity of such a calculus with the sequent system.

2.1.1. Semantic tableaux are used as refutation systems. They are built by means of a set of rules that preserve satisfiability. The *tableau expansion rules* are the following:

$$\neg\text{-rules)}\quad \frac{\neg\neg\varphi}{\varphi} \qquad \frac{\neg false}{true}$$

$$\alpha\text{-rules)}\quad \frac{\alpha_1 \wedge \alpha_2}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\neg(\alpha_1 \rightarrow \alpha_2)}{\begin{array}{c}\alpha_1\\ \neg\alpha_2\end{array}} \qquad \frac{\neg(\alpha_1 \vee \alpha_2)}{\begin{array}{c}\neg\alpha_1\\ \neg\alpha_2\end{array}}$$

$$\beta\text{-rules)}\quad \frac{\beta_1 \vee \beta_2}{\beta_1 | \beta_2} \qquad \frac{\beta_1 \rightarrow \beta_2}{\neg\beta_1 | \beta_2} \qquad \frac{\neg(\beta_1 \wedge \beta_2)}{\neg\beta_1 | \neg\beta_2}$$

If φ is a formula, a tableau for φ is a tree whose root is labelled by φ and every non-root node is obtained from a preceding node in the same branch by means of the application of an expansion rule. The following definition generalizes this notion to tableaux for sets of formulae.

Definition 3 (Tableaux) *Let $\{\varphi_1, \ldots, \varphi_s\}$ be a finite set of formulae of $\mathcal{L}_0$.*

1. *The following one branch tree is a tableau for $\{\varphi_1, \ldots, \varphi_s\}$:*

$$\begin{array}{c}\varphi_1\\ \varphi_2\\ \cdot\\ \cdot\\ \varphi_s\end{array}$$

2. *If $\mathcal{T}$ is a tableau for $\{\varphi_1, \ldots, \varphi_s\}$ and $\mathcal{T}^*$ results from $\mathcal{T}$ by the application of a tableau expansion rule, then $\mathcal{T}^*$ is a tableau for $\{\varphi_1, \ldots, \varphi_s\}$.*

Each branch of a tableau can be thought of as the conjunction of the formulae appearing in it and the whole tableau as the disjunction of its branches. A tableau branch is satisfiable if the conjunction of all the formulae labelling the branch is satisfiable. A tableau is satisfiable if one of its branches is satisfiable. The tableau system preserves satisfiability, that is, if $\mathcal{T}$ is a satisfiable tableau for Γ, then any application of a tableau expansion rule yields another satisfiable tableau.

A branch $\mathcal{B}$ of a tableau is called *closed* if both φ and $\neg\varphi$ occur in $\mathcal{B}$, for some formula φ, or if the atom $false$ occurs in $\mathcal{B}$; otherwise it is called *open*. A tableau is closed iff all its branches are closed.

Definition 4 (Refutations) *A* tableau refutation *of φ is a closed tableau for φ.*

Definition 5 (Tableaux proofs) *A* tableau proof *of φ is a closed tableau for $\neg\varphi$. φ is a theorem of the tableau system if φ has a tableau proof.*

The tableau system is sound and complete, i.e. φ is a tautology iff φ has a tableau proof.

2.1.2. A Gentzen-type calculus is a proof system given by a set of validity preserving rules. Proofs are trees labelled by *sequents*, i.e. constructs of the form $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite sets of formulae and $\Rightarrow$ is a new symbol. A sequent $\Gamma \Rightarrow \Delta$ is true in an interpretation $\mathcal{M}$ if either some formula in Γ is false in $\mathcal{M}$ or some formula in Δ is true in $\mathcal{M}$. In other terms, the sequent $\{\gamma_1, \ldots, \gamma_n\} \Rightarrow \{\delta_1, \ldots, \delta_m\}$ is interpreted as the implication $(\gamma_1 \wedge \ldots \wedge \gamma_n) \rightarrow (\delta_1 \vee \ldots \vee \delta_m)$.

The following rules are the propositional inference rules of the classical sequent calculus. In order to simplify the notation, here and in the following we shall write $\Gamma, \alpha_1, ..., \alpha_n$ instead of $\Gamma \cup \{\alpha_1, ..., \alpha_n\}$ in antecedents or consequents of sequents.

$$(\neg \Rightarrow)\ \frac{\Gamma \Rightarrow \alpha, \Delta}{\Gamma, \neg\alpha \Rightarrow \Delta} \qquad (\Rightarrow \neg)\ \frac{\Gamma, \alpha \Rightarrow \Delta}{\Gamma \Rightarrow \neg\alpha, \Delta}$$

$$(\wedge \Rightarrow)\ \frac{\Gamma, \alpha, \beta \Rightarrow \Delta}{\Gamma, \alpha \wedge \beta \Rightarrow \Delta} \qquad (\Rightarrow \wedge)\ \frac{\Gamma \Rightarrow \alpha, \Delta\ ;\ \Gamma \Rightarrow \beta, \Delta}{\Gamma \Rightarrow \alpha \wedge \beta, \Delta}$$

$$(\vee \Rightarrow)\ \frac{\Gamma, \alpha \Rightarrow \Delta\ ;\ \Gamma, \beta \Rightarrow \Delta}{\Gamma, \alpha \vee \beta \Rightarrow \Delta} \qquad (\Rightarrow \vee)\ \frac{\Gamma \Rightarrow \alpha, \beta, \Delta}{\Gamma \Rightarrow \alpha \vee \beta, \Delta}$$

$$(\rightarrow \Rightarrow)\ \frac{\Gamma \Rightarrow \alpha, \Delta\ ;\ \Gamma, \beta \Rightarrow \Delta}{\Gamma, \alpha \rightarrow \beta \Rightarrow \Delta} \qquad (\Rightarrow \rightarrow)\ \frac{\Gamma, \alpha \Rightarrow \beta, \Delta}{\Gamma \Rightarrow \alpha \rightarrow \beta, \Delta}$$

An *axiom* of the calculus is a sequent $\Gamma \Rightarrow \Delta$ such that $\Gamma \cap \Delta \neq \emptyset$.

The above rules can be read either downwards, so that they allow us to perform *deductions* and construct proofs of valid sequents starting from axioms, or upwards. In the latter case they reduce the question of the validity of a sequent to the question of the validity of one or two simpler sequents. As we shall be concerned mainly by this second reading of the rules, the calculus will be called the *reduction calculus.*

Definition 6 (Reduction tree (R-tree)) *Let Σ be a sequent. A reduction tree $\mathcal{T}$ for Σ is a finite tree whose root (endsequent) is Σ and every node Σ_i in $\mathcal{T}$ is either a leaf or it is the conclusion of an inference rule of the reduction calculus, where the premisse is (the premisses are) the node(s) in $\mathcal{T}$ immediately above Σ_i.*

Definition 7 (Sequent proof) *An R-tree $\mathcal{T}$ over the sequent Σ is a sequent proof for Σ if all its leaves are axioms.*

The sequent calculus is sound and complete: a propositional sequent is valid if and only if there exists a proof tree for it.

2.1.3. In order to recognize the fundamental identity of the two calculi, it may be useful to give a brief presentation of the original notion of tableau, dealing with signed formulae, i.e. expressions of the form $T\varphi$ or $F\varphi$, where φ is a formula [21]. The signed version of the rules given in section 2.1.1 is obtained by prefixing the formulae with T. Dual rules are immediately defined for F-formulae. For example, the following are rules of the signed tableaux system:

$$\alpha\text{-rules)}\ \frac{T(\alpha_1 \wedge \alpha_2)}{\begin{array}{c}T\alpha_1\\ T\alpha_2\end{array}} \quad \frac{F(\alpha_1 \rightarrow \alpha_2)}{\begin{array}{c}T\alpha_1\\ F\alpha_2\end{array}} \quad \frac{F(\alpha_1 \vee \alpha_2)}{\begin{array}{c}F\alpha_1\\ F\alpha_2\end{array}}$$

$$\beta\text{-rules)}\ \frac{T(\beta_1 \vee \beta_2)}{T\beta_1 | T\beta_2} \quad \frac{T(\beta_1 \rightarrow \beta_2)}{F\beta_1 | T\beta_2} \quad \frac{F(\beta_1 \wedge \beta_2)}{F\beta_1 | F\beta_2}$$

The signed systems has also rules that deals with negation:

$$\neg\text{-rules)}\ \frac{T\neg\varphi}{F\varphi} \quad \frac{F\neg\varphi}{T\varphi}$$

A branch is closed when both $T\varphi$ and $F\varphi$ occur in it. When a rule is applied to a formula, that formula is marked as used.

A tableau corresponds to a set of sequents. Every branch of the tableau corresponds to a sequent, a top sequent of an R-tree: a non-marked occurrence of a formula $T\varphi$ on the branch means that φ occurs on the left of the sequent arrow, a non-marked occurrence of $F\varphi$ means that φ occurs on the right of the sequent arrow. The tableaux defined in section 2.1.1 correspond to signed tableaux where formulae are all signed with T, i.e. their branches correspond to sequents where the right hand side is empty. The fundamental identity of the rules of the tableau and sequent systems can be easily recognized.

Given such a correspondence between the two calculi, in the sequel we shall often use ambigous expressions, such as *trees* to denote both R-trees and tableaux, and notations, such as Σ to denote both sequents and sets of sentences.

Tableaux and R-trees have properties that make them interesting from the standpoint of proof theory. They enjoy the *subformula property*: every formula added to a branch (in a sequent) is a subformula of some formula already on that branch (occurring in a lower sequent). This is essentially what gives tableau and R-trees their analytic character.

Propositional tableaux and R-trees have another property, that we call the *decomposition property*. We define a mapping from tableaux to formulae as follows: $\mathcal{F}(\mathcal{T})$ is the disjunction of the open branches of $\mathcal{T}$, where each branch $\mathcal{B}$ is identified with the conjunction of the unexpanded formulae (formulae to which no expansion rule has been applied). Then the formula ψ labelling the root of a tableaux $\mathcal{T}$ for $\{\psi\}$ is equivalent to $\mathcal{F}(\mathcal{T})$; in particular, if $\mathcal{T}$ is *acceptable* (see next subsection), $\mathcal{F}(\mathcal{T}) = DNF(\psi)$, where $DNF(\psi)$ is the reduced disjunctive normal form of φ. Analogously, if $\mathcal{T}$ is an R-tree, $\mathcal{F}(\mathcal{T})$ is the conjunction of the non-axiom leaves of $\mathcal{T}$, where each sequent $\{\gamma_1, \ldots, \gamma_n\} \Rightarrow \{\delta_1, \ldots, \delta_m\}$ is identified with the disjunction $\neg\gamma_1 \vee \ldots \vee \neg\gamma_n \vee \delta_1 \vee \ldots \vee \delta_m$. Then, the endsequent Σ of an R-tree $\mathcal{T}$ is equivalent to $\mathcal{F}(\mathcal{T})$; in particular, if $\mathcal{T}$ is acceptable, $\mathcal{F}(\mathcal{T}) = CNF(\Sigma)$, where $CNF(\Sigma)$ is the reduced conjunctive normal form of the formula corresponding to Σ.[1]

Tableaux can obviously be used also to transform a formula into CNF: the conjunction of the negation of the branches of a tableau $\mathcal{T}$ for $\{\neg\psi\}$ is $CNF(\psi)$. The same duality holds for R-trees. In fact, "tableaux are, above all, *the best way known to logicians to find a clause form (or a CNF) to a given sentence or sequent*" [1].

In propositional logic, the order of application of the rules in a tree is interchangeable (*permutability property*) and a set of formulae can have different tableaux (a sequent different R-trees) that differ only for the order of application of the rules. However, if $\mathcal{T}_1$ and $\mathcal{T}_2$ are different tableaux (R-trees) for the same (set of) formulae, then $\mathcal{F}(\mathcal{T}_1) \equiv \mathcal{F}(\mathcal{T}_2)$.

2.2 The set of abductive explanations

Let $\langle\Theta, \varphi\rangle$ be an abduction problem. Then, as $\Theta \not\models \varphi$, there is no tableau refutation for $\Theta \cup \{\neg\varphi\}$ and there is no sequent proof for $\Theta \Rightarrow \varphi$. The basic idea underlying this work is that a solution for the abduction problem $\langle\Theta, \varphi\rangle$ can be found among the formulae that force the closure of a tableau for $\Theta \cup \{\neg\varphi\}$; equivalently, among the formulae that, when added to the left hand side of the top sequents of an R-tree for $\Theta \Rightarrow \varphi$, make them all valid. Of course, it would be meaningless to close an unexpanded tableaux by adding the atom $false$ or the formula φ. So, the trees have to be developed as far as possible and a careful choice of the literals that close each branch has to be performed. In the sequel, we define some notions that are useful in this respect. Here and in the following, tableaux for $\Theta \cup \{\neg\varphi\}$ and R-trees above $\Theta \Rightarrow \varphi$ will be generically called trees for the abduction problem $\langle\Theta, \varphi\rangle$.

Definition 8 (Acceptable trees) *Let $\mathcal{T}$ be a tableau. A branch $\mathcal{B}$ of $\mathcal{T}$ is* fundamental *if each non literal formula occurring in $\mathcal{B}$ has been expanded with the appropriate expansion rule. $\mathcal{T}$ is* acceptable *if all its branches are fundamental.*

Let $\mathcal{T}$ be an R-tree. A sequent Σ of $\mathcal{T}$ is fundamental *iff it contains only atoms. $\mathcal{T}$ is* acceptable *if all its leaves are fundamental.*

[1] The decomposition and subformula properties hold for the above formulations of the calculi, that make no use of structural rules.

Definition 9 (Threads and closing sets) *Let $\mathcal{B}$ be a fundamental branch of a tableau. The thread associated to $\mathcal{B}$, $\rho(\mathcal{B})$ is the set of the literals labeling nodes of $\mathcal{B}$. The closing set for $\mathcal{B}$ is $\tau(\mathcal{B}) = \{\neg\lambda \mid \lambda \in \rho(\mathcal{B})\}$, where $\neg\lambda$ is the complement of λ.*

If $\mathcal{T}$ is an acceptable tableau, the set $\mathcal{S}(\mathcal{T})$ of the minimal closing sets of (the open branches of) $\mathcal{T}$ is $\{\tau(\mathcal{B}) \mid \mathcal{B}$ is an open branch in $\mathcal{T}$ and there is no branch $\mathcal{B}'$ in $\mathcal{T}$ such that $\tau(\mathcal{B}') \subset \tau(\mathcal{B})\}$.

Let $\Sigma = \Gamma \Rightarrow \Delta$ be a fundamental sequent. The thread associated to Σ, $\rho(\Sigma)$ is $\Gamma \cup \{\neg\lambda \mid \lambda \in \Delta\}$. The closing set for Σ is $\tau(\Sigma) = \{\neg\lambda \mid \lambda \in \rho(\Sigma)\} = \{\neg\lambda \mid \lambda \in \Gamma\} \cup \Delta$.

If $\mathcal{T}$ is an acceptable R-tree, the set $\mathcal{S}(\mathcal{T})$ of the minimal closing sets of (the non axiom leaves of) $\mathcal{T}$ is $\{\tau(\Sigma) \mid \Sigma$ is a non axiom leaf in $\mathcal{T}$ and there is no leaf Σ' in $\mathcal{T}$ such that $\tau(\Sigma') \subset \tau(\Sigma)\}$.

$\mathcal{S}(\mathcal{T})$ will sometimes be briefly called the set of the closing sets for $\mathcal{T}$, but it has to be noted that it collects the *subset-minimal* closing sets of the *open* branches (*non-axiom* leaves) of $\mathcal{T}$.

As a consequence of the decomposition property, if $\mathcal{T}_1$ e $\mathcal{T}_2$ are acceptable trees for the same set of formulae or sequent, then $\mathcal{S}(\mathcal{T}_1) = \mathcal{S}(\mathcal{T}_2)$; therefore, if Σ is a set of propostional formulae (a propositional sequent), the set $\mathcal{S}(\Sigma)$ of the minimal closing sets of the open branches (non axiom leaves) of any tableau (R-tree) for Σ is well defined.

Definition 10 (Closures) *Let $\mathcal{T}$ be an acceptable tableau (an acceptable R-tree) for the abduction problem $\langle\Theta, \psi\rangle$ and $\mathcal{S}(\mathcal{T}) = \{\tau_1, \ldots, \tau_n\}$ the set of the closing sets of $\mathcal{T}$. Let g be any choice function for the elements of $\mathcal{S}(\mathcal{T})$, i.e. $g(\tau_i) \in \tau_i$.*

(i) If $\mathcal{S}(\mathcal{T}) = \emptyset$, $\mathcal{T}$ has a single closure, the atom true. *(This is the case where $\Theta \models \psi$).*

(ii) If $\mathcal{S}(\mathcal{T}) \neq \emptyset$ and for any choice function g for the elements of $\mathcal{S}(\mathcal{T})$ the set $\{g(\tau_1), \ldots, g(\tau_n)\}$ contains a pair of complementary literals, then the only closure for $\mathcal{T}$ is the atom false. *(This is the case where $\Theta \models \neg\psi$).*

(iii) Otherwise $\alpha = g(\tau_1) \wedge \ldots \wedge g(\tau_n)$ is a closure for $\mathcal{T}$ iff α does not contain a pair of complementary literals.

In the above definition, it is intended that α is the conjunction of the literals in the *set* $\{g(\tau_1) \wedge \ldots \wedge g(\tau_n)\}$, i.e. literals are not repeated.

If $\mathcal{T}$ is any acceptable tableau or R-tree, we denote by $\mathcal{E}(\mathcal{T})$ the set of all the closures of $\mathcal{T}$. Clearly, different acceptable trees (tableaux or R-trees) for the same abduction problem $\langle\Theta, \varphi\rangle$ have the same closures. Let then:

$$\mathcal{E}(\Theta, \varphi) = \mathcal{E}(\mathcal{T}) \text{ for any tree } \mathcal{T} \text{ for } \langle\Theta, \varphi\rangle$$

The following theorem establishes completeness and a form of soundness of the tableau/sequent based procedure that solves abduction problems by generation of the set $\mathcal{E}(\Theta, \varphi)$.

Theorem 1 *Let $\langle \Theta, \varphi \rangle$ be an abduction problem. Then every element of $\mathcal{E}(\Theta, \varphi)$ is a non-contradictory explanation for $\langle \Theta, \varphi \rangle$* (soundness) *and any minimal and non-contradictory explanation for $\langle \Theta, \varphi \rangle$ is an element of $\mathcal{E}(\Theta, \varphi)$* (completeness).

Clearly, the set of all but only the minimal and Θ-consistent explanations for $\langle \Theta, \varphi \rangle$ can be obtained by filtering the set $\mathcal{E}(\Theta, \varphi)$ and retaining only its minimal elements that are consistent with Θ. In the next subsection we show how to perform the consistency test, fundamentally without any extra effort, and give a characterization of minimal explanations that corresponds to a non-deterministic algorithm for the construction of a single minimal and Θ-consistent explanation for $\langle \Theta, \varphi \rangle$.

2.3 Minimality and consistency with the theory

2.3.1. A straightforward consequence of Theorem 1 is that the set of the minimal explanations for $\langle \Theta, \varphi \rangle$ is equal to the set $min(\mathcal{E}(\Theta, \varphi))$ of the minimal closures for any tree $\mathcal{T}$ for $\langle \Theta, \varphi \rangle$, i.e.

$min(\{\alpha \mid \text{for all } \tau \in \mathcal{S}(\mathcal{T}), (\alpha \cap \tau) \neq \emptyset\})$.

where, if Γ is a set of formulae, $min(\Gamma)$ denotes its subset containing only minimal elements w.r.t. $\models$.

This result leads to the construction of the set of minimal explanations of $\langle \Theta, \varphi \rangle$, but it does not help to build minimal explanations one by one, because in order to test their minimality each of them has to be compared with all the other elements of $\mathcal{E}(\Theta, \varphi)$. We are now going to show how this can be done, using only the set of the closing sets of any tree for $\langle \Theta, \varphi \rangle$.

Definition 11 *Let $\langle \tau_1, \ldots, \tau_m \rangle$ be any ordered and non-empty set of sets of literals and α a conjunction of literals. Then α is* minimally determined *by $\langle \tau_1, \ldots, \tau_m \rangle$ iff one of the following conditions holds:*

(a) $m = 1$ *and* $\alpha \in \tau_1$*;*

(b) $m > 1$*, α is minimally determined by $\langle \tau_1, \ldots, \tau_{m-1} \rangle$ and* $\alpha \cap \tau_m \neq \emptyset$*;*

(c) $m > 1$*, condition (b) does not hold and there exists* $\lambda \in \alpha \cap (\tau_m - (\tau_1 \cup \ldots \cup \tau_{m-1}))$ *such that, if $\alpha \equiv \gamma \wedge \lambda$, then $\neg\lambda \notin \gamma$ and γ is minimally determined by $\langle \tau_1, \ldots, \tau_{m-1} \rangle$.*

The condition (c) of Definition 11 ensures the soundness of the procedure (i.e. when $\mathcal{T}$ is a tree for an abductive problem and $\tau_1, \ldots, \tau_m$ are the closing set in $\mathcal{S}(\mathcal{T})$, then only minimal explanation are generated) but completeness is lost, unless any permutation of the sets $\tau_1, \ldots, \tau_m$ is considered. Next definition copes with this fact.

Definition 12 *Let $\tau_1, \ldots, \tau_m$ be sets of literals and α a conjunction of literals. Then α is* minimally determined *by $\{\tau_1, \ldots, \tau_m\}$ iff there exists a permutation $\langle \tau_{p_1}, \ldots, \tau_{p_m} \rangle$ of $\tau_1, \ldots, \tau_m$, such that α is minimally determined by $\langle \tau_{p_1}, \ldots, \tau_{p_m} \rangle$.*

Lemma 1 *Let $\mathcal{T}$ be a tree with non-empty $\mathcal{S}(\mathcal{T})$ and such that $false$ is not a minimal closure of $\mathcal{T}$. Then, a C-formula α is a minimal closure for $\mathcal{T}$ iff α is minimally determined by $\mathcal{S}(\mathcal{T})$.*

2.3.2. If Θ is any set of consistent formulae and $\mathcal{T}$ is any tableau for Θ or $\mathcal{T}$ is an R-tree for $(\Theta \Rightarrow\ )$, then $\mathcal{T}$ will be generically called an *analytic tree* for Θ and $\mathcal{E}(\Theta)$ will denote $\mathcal{E}(\mathcal{T})$, i.e. $\mathcal{E}(\Theta, false)$.

In order to obtain from $\mathcal{E}(\Theta, \varphi)$ the explanations for $\langle \Theta, \varphi \rangle$ that are consistent with Θ, we note that the set of all the minimal and Θ-consistent explanations for $\langle \Theta, \varphi \rangle$ is $min(\mathcal{E}(\Theta, \varphi)) - min(\mathcal{E}(\Theta))$. Thus, we can first generate all the minimal explanations for the negation of the theory (i.e. the set $min(\mathcal{E}(\Theta))$ and then remove them from $min(\mathcal{E}(\Theta, \varphi))$. As remarked in [12], this is reasonable, if we assume that the theory Θ is not frequently modified, so that the set $min(\mathcal{E}(\Theta))$ can be stored once and for all and used each time an abduction problem has to be solved.

However, the subformula property enjoyed by the tableau/sequent calculi can help do something more. The formulation of the systems can be modified so that the origin of every formula as a subformula of Θ or a subformula of φ can be recognized. In the case of the sequent calculus, for example, the rules can be modified so that, in the sequents of a deduction, formulae never change sides. The rules $(\neg \Rightarrow\ )$, $(\ \Rightarrow \neg)$, $(\rightarrow \Rightarrow\ )$ and $(\ \Rightarrow \rightarrow)$ are to be dropped and the following rules are added:

$$(\neg\wedge \Rightarrow)\ \frac{\Gamma, \neg\alpha \Rightarrow \Delta\ ;\ \Gamma, \neg\beta \Rightarrow \Delta}{\Gamma, \neg(\alpha \wedge \beta) \Rightarrow \Delta} \qquad (\Rightarrow \neg\wedge)\ \frac{\Gamma \Rightarrow \neg\alpha, \neg\beta, \Delta}{\Gamma \Rightarrow \neg(\alpha \wedge \beta), \Delta}$$

$$(\neg\vee \Rightarrow)\ \frac{\Gamma, \neg\alpha, \neg\beta \Rightarrow \Delta}{\Gamma, \neg(\alpha \vee \beta) \Rightarrow \Delta} \qquad (\Rightarrow \neg\vee)\ \frac{\Gamma \Rightarrow \neg\alpha, \Delta\ ;\ \Gamma \Rightarrow \neg\beta, \Delta}{\Gamma \Rightarrow \neg(\alpha \vee \beta), \Delta}$$

$$(\rightarrow \Rightarrow)\ \frac{\Gamma, \neg\alpha \Rightarrow \Delta\ ;\ \Gamma, \beta \Rightarrow \Delta}{\Gamma, \alpha\rightarrow\beta \Rightarrow \Delta} \qquad (\Rightarrow \rightarrow)\ \frac{\Gamma \Rightarrow \neg\alpha, \beta, \Delta}{\Gamma \Rightarrow \alpha\rightarrow\beta, \Delta}$$

$$(\neg\rightarrow \Rightarrow)\ \frac{\Gamma, \alpha, \neg\beta \Rightarrow \Delta}{\Gamma, \neg(\alpha\rightarrow\beta) \Rightarrow \Delta} \qquad (\Rightarrow \neg\rightarrow)\ \frac{\Gamma \Rightarrow \alpha, \Delta\ ;\ \Gamma \Rightarrow \neg\beta, \Delta}{\Gamma \Rightarrow \neg(\alpha\rightarrow\beta), \Delta}$$

A sequent $\Gamma \Rightarrow \Delta$ in this calculus is an axiom if and only if either $\Gamma \cap \Delta \neq \emptyset$, or Γ contains a pair of complementary formulae, or Δ contains a pair of complementary formulae.

The leaves $\Gamma_i \Rightarrow \Delta_i$ of an acceptable R-tree for a sequent $\Theta \Rightarrow \varphi$ have the property that every literal in Γ_i is an ancestor of a subformula in Θ and every literal in Δ_i is an ancestor of a subformula in φ. Such a feature, i.e. *analicity* of R-trees (and tableaux), has no correspondence in resolution proofs.

If $(\Lambda_1 \Rightarrow \Psi_1), \dots, (\Lambda_m \Rightarrow \Psi_m)$ are all the axiom leaves in an R-tree for $(\Theta \Rightarrow \varphi)$ and $(\Gamma_1 \Rightarrow \Delta_1), \dots, (\Gamma_n \Rightarrow \Delta_n)$ are the non-axiom leaves, it follows that the leaves of an R-tree $\mathcal{T}$ for $(\Theta \Rightarrow\ )$ are exactly $(\Lambda_1 \Rightarrow\ ), \dots, (\Lambda_m \Rightarrow\ )$, $(\Gamma_1 \Rightarrow\ ), \dots, (\Gamma_n \Rightarrow\ )$. The non-axiom leaves of $\mathcal{T}$ include certainly $(\Gamma_1 \Rightarrow\ )$, $\dots, (\Gamma_n \Rightarrow\ )$, but possibly also some of the $(\Lambda_i \Rightarrow\ )$, let us say $(\Lambda_1 \Rightarrow\ ), \dots,$ $(\Lambda_k \Rightarrow\ )$ for some $k \leq m$. To simplify the notation, let us rename $(\Lambda_i \Rightarrow\ )$ as $(\Gamma_{n+i} \Rightarrow\ )$ for $i = 1, ..., k$.

If a C-formula σ is an element of $\mathcal{E}(\Theta)$ then σ is a closure for $\mathcal{T}$, and for all $\tau \in \mathcal{S}(\mathcal{T})$, $\tau \cap \sigma \neq \emptyset$. Therefore, an element of $min(\mathcal{E}(\Theta, \varphi))$ is consistent with Θ iff for some $\tau \in \mathcal{S}(\mathcal{T})$, $\tau \cap \sigma = \emptyset$.

Note that the sequents $(\Gamma_1 \Rightarrow\ ) \dots, (\Gamma_{n+k} \Rightarrow\ )$, and consequently the sets in $\mathcal{S}(\mathcal{T})$, can be built after the construction of the R-tree for $\Theta \Rightarrow \varphi$ with a minimal effort. The method does not require the explicit construction of $min(\mathcal{E}(\Theta))$.

The same mechanism can be adapted to the tableau-based method. For example, by the permutability of the rules in the propositional case, the tableau can be built by first applying the expansion rules to Θ as far as possible, collecting the threads deriving from Θ and finally expanding $\neg\varphi$. Alternatively, mimicking the sequent-based method, threads can be thought of as pairs $\langle \tau^{\Theta}, \tau^{\neg\varphi} \rangle$, where the elements of τ^{Θ} are subformulae of Θ and the elements of $\tau^{\neg\varphi}$ are subformulae of $\neg\varphi$.

2.3.3. Basing on the above observations and the results in 2.3.1, here follows a characterization of minimal and Θ-consistent explanations, that does not refer to the set of all the explanations for the abduction problem, but makes use of the collection of closing sets for any analytic tree for it. Such a characterization is equivalent to the usual definition. On its basis, an algorithm can easily be defined, that non deterministically builds a single explanation for $\langle \Theta, \varphi \rangle$ that is minimal and consistent with Θ.

If $\mathcal{T}$ is any tree for $\langle \Theta, \varphi \rangle$, let $\mathcal{T}'$ be the subtree of $\mathcal{T}$ where no rule is applied to $\neg\varphi$ (φ). The characterization of Θ-consistent explanations for $\langle \Theta, \varphi \rangle$ refers to the two sets of closing sets $\mathcal{S}(\mathcal{T})$ and $\mathcal{S}(\mathcal{T}')$ and is justified by the following lemma.

Lemma 2 *Let $\mathcal{T}$ be any analytic tree for Θ. Then a C-formula α is consistent with Θ iff there exists $\tau \in \mathcal{S}(\mathcal{T})$ such that $\alpha \cap \tau = \emptyset$.*

In fact, if α is consistent with Θ, then there is some branch in any tree for Θ that is not closed by adding (and expanding) α. Then there is also a minimal branch that is not closed by α. So, for some $\tau \in \mathcal{S}(\mathcal{T})$ it must be that $\tau \cap \alpha = \emptyset$.

Lemmas 1 and 2 merge in the following:

Theorem 2 *Let $\mathcal{T}$ be any tree for $\langle\Theta, \varphi\rangle$ and $\mathcal{T}'$ the subtree of $\mathcal{T}$ where no rule is applied to $\neg\varphi$ (φ). Then a C-formula α is a minimal and Θ-consistent explanation for $\langle\Theta, \varphi\rangle$ iff α is minimally determined by $\mathcal{S}(\mathcal{T})$ and there exists $\tau \in \mathcal{S}(\mathcal{T}')$ such that $\alpha \cap \tau = \emptyset$.*

The above theorem clearly gives a way to define a sound and complete non-deterministic algorithm for the construction of a single explanation for an abduction problem.

3 Full first order abduction

3.1 Automated theorem proving in first order sequent calculus

The first order rules for the universal quantifier in sequent calculus are usually given the following formulation, where t is any term and a a free variable not occurring in $\Gamma, \forall x\alpha(x) \Rightarrow \Delta$:

$$(\forall \Rightarrow)\ \frac{\Gamma, \alpha(t) \Rightarrow \Delta}{\Gamma, \forall x\alpha(x) \Rightarrow \Delta} \qquad (\Rightarrow \forall)\ \frac{\Gamma \Rightarrow \alpha(a), \Delta}{\Gamma \Rightarrow \forall x\alpha(x), \Delta}$$

These rules preserve validity. However, if we need falsity preserving rules, so that validity is preserved in the calculus also when the rules are read upwards, a different version of $(\forall \Rightarrow)$ has to be adopted:

$$(\forall \Rightarrow)\ \frac{\Gamma, \alpha(t), \forall x\alpha(x) \Rightarrow \Delta}{\Gamma, \forall x\alpha(x) \Rightarrow \Delta}$$

The main problem in automatizing proof search in this calculus is the choice of the term t in such a rule. Sometimes the rule has been stated as follows [11, 7]:

$$(\forall \Rightarrow)\ \frac{\Gamma, \alpha(t_1), \ldots, \alpha(t_k), \forall x\alpha(x) \Rightarrow \Delta}{\Gamma, \forall x\alpha(x) \Rightarrow \Delta}$$

where $t_1, \ldots, t_k$ are terms from the language up to a given depth. This allows the mechanization of the calculus to perform validity checking.

In [10, 2] it is proposed to delay the choice of the terms until an attempt to unify a pair of formulae occurring in the opposite sides of the same sequent succeeds. The notion of *metavariable* or *dummy* is introduced and the following version of the rule is given, that we shall also adopt:

$$(\forall \Rightarrow)\qquad \frac{\Gamma, \alpha(d_i), \forall x\alpha(x) \Rightarrow \Delta}{\Gamma, \forall x\alpha(x) \Rightarrow \Delta}$$
where d_i is a new metavariable

When the rule is used in a reduction calculus, it is meant to build not a proper derivation or proof, but just a skeleton that can be changed into a derivation or a proof by application of a substitution of terms for dummies. The problem is that the application of substitutions does not preserve the correcness of $(\Rightarrow \forall)$ inferences, because the *eigenvariable* conditions may be violated.

Along the lines of Fitting [6], that defines a tableau calculus handling also dynamic skolemization (see next section), it has been proposed in [20] to modify the formulation of the $(\Rightarrow \forall)$ rule by using Herbrand (or Skolem) functions, so that application of substitutions preserves correctness:

$$(\Rightarrow \forall) \quad \frac{\Gamma \Rightarrow \alpha(h_i(d_0, ..., d_n)), \Delta}{\Gamma \Rightarrow \forall x \alpha(x), \Delta}$$

where h_i is a new function and $d_0, ..., d_n$ are all the metavariables occurring in $\Gamma, \forall x \alpha(x) \Rightarrow \Delta$. The calculus defined in [20] - that deals with intuitionistic logic - is actually more sophisticated, in that a list of "dominating" dummies is attached to every formula in a sequent, that augments with every application of $(\forall \Rightarrow)$ and determines the set of variables that have to occur in an h-term introduced by an application of $(\Rightarrow \forall)$. However, in order to simplify notation, exposition and proofs, we shall not deal with such an improvement in this work.

3.2 Free variable semantic tableaux and R-trees

In the following the main formal definitions, both for the case of free-variable semantic tableaux [6] and for the sequent calculus, are introduced.

Let $\mathcal{L}$ be a first order language extending $\mathcal{L}_0$. Terms, formulae, literals, free and bound occurences of variables in a formula are defined as usual. $\mathcal{L}^{sko}$ is the extension of $\mathcal{L}$, obtained by adding a distinct (countable) set of new symbols, the *metavariables* (or *dummy variables*) $d_0, d_1, d_2 \ldots$ and, for any n, a (countable) set of new n-place function symbols, called *Skolem functions* or *h-functions*[2] $h_0^n, h_1^n, \ldots$ (the superscript will often be omitted).

The first order tableau rules include the propositional expansion rules and the following rules, where d is a new metavariable that does not occur elsewhere in the tableau, h is a new Skolem function, and $d_1, \ldots, d_n$ are all the metavariables occurring in the branch:

$$(\gamma - rules) \quad \frac{\forall x \gamma(x)}{\gamma(d)} \qquad \frac{\neg \exists x \gamma(x)}{\neg \gamma(d)}$$

$$(\delta - rules) \quad \frac{\exists x \delta(x)}{\delta(h(d_1, \ldots, d_n))} \quad \frac{\neg \forall x \delta(x)}{\neg \delta(h(d_1, \ldots, d_n))}$$

The reduction calculus we are adopting contains the following quantifier rules, where d is a new metavariable that does not occur elsewhere in the R-tree,

[2]The reason for this double naming is that these functional symbols are used as Skolem functions in the tableau (refutation) system, while they rather appear as Herbrand functions (h-functions) in the sequent (validation) calculus.

h is a new Skolem function, and $d_1, \ldots, d_n$ are all the metavariables occurring in $\Gamma, \forall x\alpha(x), \Delta$.

$$(\forall \Rightarrow\ )\ \frac{\Gamma, \alpha(d), \forall x\alpha(x) \Rightarrow \Delta}{\Gamma, \forall x\alpha(x) \Rightarrow \Delta} \qquad (\ \Rightarrow \forall)\ \frac{\Gamma \Rightarrow \alpha(h(d_0, ..., d_n)), \Delta}{\Gamma \Rightarrow \forall x\alpha(x), \Delta}$$

$$(\exists \Rightarrow\ )\ \frac{\Gamma, \alpha(h(d_0, ..., d_n)) \Rightarrow \Delta}{\Gamma, \exists x\alpha(x) \Rightarrow \Delta} \qquad (\ \Rightarrow \exists)\ \frac{\Gamma \Rightarrow \alpha(d), \exists x\alpha(x), \Delta}{\Gamma \Rightarrow \exists x\alpha(x), \Delta}$$

The *unrestricted* rules, i.e. $(\forall \Rightarrow\ )$ and $(\ \Rightarrow \exists)$ correspond to the γ-rules for tableaux, while $(\exists \Rightarrow\ )$ and $(\ \Rightarrow \forall)$ correspond to δ-rules. Note that, because of the δ-rules, the decomposition property does not hold.

Clearly, as γ-rules may be applied infinitely many times, first order trees may be infinite. However, the definitions of R-trees and tableaux are extended to first order so that only finite trees are considered.

Definition 13 (Uninstantiated tableau and reduction tree (U-tree)) *Let Σ be a set of formulae. An uninstantiated tableau $\mathcal{T}$ for Σ is a* finite *tree that is either a one-branch tableau for Σ or it is obtained from a tableau for Σ by application of first order expansion rules.*

Let Σ be a sequent. An uninstantiated reduction tree $\mathcal{T}$ over Σ is a finite tree whose root (endsequent) is Σ and every node Σ_i in $\mathcal{T}$ is either a leaf or it is the conclusion of an inference rule of the reduction calculus, where the premisse is (the premisses are) the node(s) in $\mathcal{T}$ immediately above Σ_i.

Substitutions are defined as usual. They are intended to affect only metavariables, that are all distinct from bound variables, so that substitutions are always free.

Definition 14 (Instantiated tree (I-tree)) Π *is an instantiated tree (either a tableau or an R-tree) if it is obtained by applying a substitution θ of terms for metavariables to every formula of a U-tree. I.e., for every U-tree $\mathcal{T}$ and substitution θ, $\mathcal{T}\theta$ is an I-tree.*

Definition 15 (Refutations and proofs) *An instantiated tableau Π for Σ is a refutation of Σ if it is closed. Π is a proof of φ if Π is a refutation of $\neg\varphi$.*

An instantiated reduction tree Π for Σ is a proof of Σ if all its leaves are axioms.

The free variable semantic tableau system is sound and complete [6], and so is the free variable sequent system.

Let Σ be a set of formulae (a sequent). In the general first order case, a possibly infinite collection of U-trees can be constructed for Σ: $\mathcal{T}_1, \mathcal{T}_2, \ldots$. Two of them may differ either on the order of application of the rules of the calculus, or because one is the extension of the other, or both (they are extensions of two other U-trees that are permutations one of the other). In fact, in spite of the reusability of γ-rules, δ and γ expansion rules are not interchangeable.

Moreover, every U-tree $\mathcal{T}_i$ for Σ may correspond to different I-trees, one for each substitution applicable to $\mathcal{T}_i$.

For the purposes of both proof search and abduction, the number of the trees that have to be considered can be reduced. For example, completeness is not lost if we admit only substitutions that are generated as most general unifiers of formulas occurring both positive and negative in the same branch (sequent). Although this kind of results is surely important, we shall not deal with this question in this work. The conceptual point is in fact that infinity cannot be ruled out of first order logic. However, the problems that arise with the undecidability of the predicate calculus mainly reflect upon the preference criteria of consistency with the theory (obviously) and minimality. In fact, not only is implication between first-order C-formulae undecidable [19], but also a first order abduction problem can have an infinite number of explanations, so, in general, we cannot determine whether a given explanation is minimal just by comparison with the others. There are even cases where no explanation is minimal [12].

These observations suggest that both criteria of consistency with the theory and minimality should be somehow relaxed. This point is not developed further here, but we define an abductive method where only a small part of the non-minimal explanations are recognized and rejected. W.r.t. propositional abduction, a new font of non-determinism stems from the choice of a single finite U-tree for the abduction problem (i.e. we stop at any stage of the development of a possibly infinite tree) and the choice of a substitution for it. On the basis of the resulting I-tree, a set of explanations for the original problem is built.

As a final remark, we note that in first order logic it is undecidable even to detemine whether a pair $\langle \Theta, \varphi \rangle$ is a genuine abduction problem. Consequently, any method for performing abduction should not rely on the assumptions that $\Theta \models \varphi$ and $\Theta \not\models \neg\varphi$.

The following definitions determine when a tree is acceptable for abduction, i.e. when its construction can be interrupted in order to generate a set of explanations.

Definition 16 (Acceptable U-tree (AU-tree)) *A branch in a tableau is* **fundamental** *if every non literal occurrence in the branch has been applied the appropriate expansion rule at least once.*

A sequent is fundamental iff it only contains either atoms or formulas of the form $\exists x\alpha$ in the consequent or $\forall x\alpha$ in the antecedent.

An U-tree is **acceptable** *if all its branches (leaves) are fundamental.*

Definition 17 (Acceptable I-tree (AI-tree)) *An I-tree $\Pi = \mathcal{T}\theta$ is acceptable iff $\mathcal{T}$ is an AU-tree.*

For each of the (finite) AI-trees Π for an abduction problem $\langle\Theta,\varphi\rangle$, we show how to build a finite set of explanations $\mathcal{E}^{FOL}(\Pi)$ for $\langle\Theta,\varphi\rangle$.

Soundness will amount to saying that if a given formula α is an element of $\mathcal{E}^{FOL}(\Pi)$, then α is not contradictory and $\Theta\cup\{\alpha\}\models\varphi$. However, α is not necessarily a minimal explanation. In particular, it may be the case that $\Theta\models\varphi$ even if non valid explanations are generated.

Completeness will guarantee that, for any non-contradictory formula α such that $\Theta\cup\{\alpha\}\models\varphi$, there exists an AI-tree Π for $\langle\Theta,\varphi\rangle$ such that $\mathcal{E}^{FOL}(\Pi)$ contains a logical consequence γ of α.

3.3 First order explanations

The notions of thread, closing set and closure for acceptable I-trees Π are immediate extensions of the corresponding propositional ones. Note however that, in the case of sequents, only literals are collected in threads and closing sets. If Π is an acceptable I-tree, then $\mathcal{E}(\Pi)$ denotes, as before, the set of all the closures of Π.

Now, an element of $\mathcal{E}(\Pi)$ is in the language $\mathcal{L}^{sko}\supseteq\mathcal{L}$, so it may not be an explanation for $\langle\Theta,\varphi\rangle$. In order to obtain explanations, all the metavariables and the skolem terms introduced in the tree for $\langle\Theta,\varphi\rangle$ are to be replaced by suitably quantified variables.

Definition 18 (Reverse skolemization) [3] *Let α be a formula in $\mathcal{L}^{sko}$ and let $st(\alpha)=\{t_1,\ldots,t_k\}$ be the set of the terms occurring in α that are not in $\mathcal{L}$. Let $\langle t_{p_1},\ldots,t_{p_k}\rangle$ be any total ordering of the elements in $st(\alpha)$ such that for all i and j, if t_{p_i} properly occurs in t_{p_j} then $i<j$. Then $Qx_1\ldots Qx_k\alpha'$ is obtained from α by reverse skolemization on the basis of $\langle t_{p_1},\ldots,t_{p_k}\rangle$ iff*

- *α' is obtained from α by replacing each term t_{p_i} with the (new) variable x_i;*
- *for all i, if t_{p_i} is a metavariable, then x_i is existentially quantified, otherwise, if t_{p_i} is a skolem term, then x_i is universally quantified.*

The set of all the formulae obtainable by reverse skolemization from α will be denoted by $desk(\alpha)$.

The fact that $desk(\alpha)$ is not necessarily a singleton is illustrated by the sample case where $\alpha=p(d_1,h_1(d_1),d_2,h_2(d_2))$; in fact the two elements of $desk(\alpha)=\{\exists x_1\forall x_2\exists x_3\forall x_4 p(x_1,x_2,x_3,x_4),\exists x_1\forall x_2\exists x_3\forall x_4 p(x_3,x_4,x_1,x_2)\}$ are not equivalent.

An algorithm for reverse skolemization can be found in [4].

[3] Although we are using the term *skolemization*, that is more familiar to the AI community, what is actually defined is reverse *herbrandization*.

Definition 19 (First order closures) *Let Π be any I-tree for the abduction problem $\langle\Theta, \varphi\rangle$ and $\mathcal{E}(\Pi)$ the set of all the closures of Π. Then the set of the first order closures of Π is*

$$\mathcal{E}^{FOL}(\Pi) = \{\alpha \mid \alpha \in min(desk(\gamma)) \textit{ for some } \gamma \in min(\mathcal{E}(\Pi))\}.$$

If $\langle\Theta, \varphi\rangle$ is an abduction problem and
$\Re = \{\Pi \mid \Pi$ *is an acceptable I-tree for* $\langle\Theta, \varphi\rangle\}$,
then

$$\mathcal{E}^{FOL}(\Theta, \varphi) = \bigcup_{\Pi \in \Re} \mathcal{E}^{FOL}(\Pi)$$

The definition of $\mathcal{E}^{FOL}(\Theta, \varphi)$ is quite naive. Investigating the relations that hold between the sets $\mathcal{E}^{FOL}(\Pi_1)$ and $\mathcal{E}^{FOL}(\Pi_2)$ when Π_1 is an expansion of Π_2 would lead to a better characterization of $\mathcal{E}^{FOL}(\Theta, \varphi)$.

The following theorem establishes that the abduction calculus that generates elements of $\mathcal{E}^{FOL}(\Theta, \varphi)$ is sound and complete.

Theorem 3 *Let $\langle\Theta, \varphi\rangle$ be an abduction problem. Then for any element α of $\mathcal{E}^{FOL}(\Theta, \varphi)$, $\Theta \cup \{\alpha\} \models \varphi$ (*soundness*). If γ is a consistent C-formula such that $\Theta \cup \{\gamma\} \models \varphi$, then there exists an element α of $\mathcal{E}^{FOL}(\Theta, \varphi)$ such that $\gamma \models \alpha$ (*completeness*).*

Example 1 Let Θ contain the single formula $\forall x \forall y \forall z (p(x, y) \wedge p(y, z) \rightarrow p(x, z))$ and $\varphi = \exists x p(x, x)$. This is the exampled used in [12] to show that in the first order case there may be no minimal explanations. An unistantiated tableau for $\Theta \cup \{\neg\varphi\}$ is the following tree $\mathcal{T}$:

(1) $\forall x \forall y \forall z (p(x,y) \wedge p(y,z) \rightarrow p(x,z))$

(2) $\neg \exists x p(x,x)$

$\vdots$

(3) $p(d_1,d_2) \wedge p(d_2,d_3) \rightarrow p(d_1,d_3)$ (from 1)

(4) $\neg p(d_4,d_4)$ (from 2)

(5) $\neg(p(d_1,d_2) \wedge p(d_2,d_3)$ (from 3)

(6) $p(d_1,d_3)$

(7) $\neg p(d_1,d_2)$

(8) $\neg p(d_2,d_3)$

The three threads of $\mathcal{T}$ are $\rho_1 = \{\neg p(d_4,d_4), \neg p(d_1,d_2)\}$, $\rho_2 = \{\neg p(d_4,d_4), \neg p(d_2,d_3)\}$ and $\rho_3 = \{\neg p(d_4,d_4), p(d_1,d_3)\}$. If the substitution $\theta = \{d_1/d_4, d_1/d_3\}$ is applied to $\mathcal{T}$, then ρ_3 closes and the set $\mathcal{S}(\mathcal{T}\theta)$ contains the two following closing sets:

$\tau_1 = \{p(d_1,d_1), p(d_1,d_2)\}$,

$\tau_2 = \{p(d_1,d_1), p(d_2,d_1)\}$.

The corresponding minimal closures are $p(d_1,d_1)$ and $p(d_1,d_2) \wedge p(d_2,d_1)$, that, by deskolemization, generate the explanations $\exists x p(x,x)$ (trivial) and $\exists x \exists y (p(x,y) \wedge p(y,x))$. Clearly, the tree $\mathcal{T}$ can be further expanded and new explanations generated.

Example 2 The following simple example shows the use of herbrand functions in the sequent case. Let $\Theta = \{\forall x(\forall y p(x,y) \rightarrow q(x))\}$ and $\varphi = \forall x q(x)$. The following derivation tree $\mathcal{T}$ is an U-tree for $\Theta \Rightarrow \varphi$.

$$\dfrac{\dfrac{\dfrac{\forall x(\forall y p(x,y)\rightarrow q(x)) \Rightarrow q(h_0), p(d, h_1(d))}{\forall x(\forall y p(x,y)\rightarrow q(x)) \Rightarrow q(h_0), \forall y p(d,y)} \;;\; \forall x(\forall y p(x,y)\rightarrow q(x)), q(d) \Rightarrow q(h_0)}{\forall x(\forall y p(x,y)\rightarrow q(x)), \forall y p(d,y)\rightarrow q(d) \Rightarrow q(h_0)}}{\dfrac{\forall x(\forall y p(x,y)\rightarrow q(x)) \Rightarrow q(h_0)}{\forall x(\forall y p(x,y)\rightarrow q(x)) \Rightarrow \forall x q(x)}}$$

Applying the substitution $\theta = \{h_0/d\}$ makes the leaf

$\forall x(\forall y p(x,y)\rightarrow q(x)), q(d) \Rightarrow q(h_0)$

an axiom. So, the only closing set of $\mathcal{T}\theta$ is $\tau = \{q(h_0), p(h_0, h_1(h_0))\}$. Consequently, $\mathcal{E}^{FOL}(\mathcal{T}\theta) = \{\forall x q(x), \forall x \forall y p(x,y)\}$.

4 Concluding remarks

The analysis of some of the still open problems in the formalization of abductive reasoning by means of sequent/tableau calculi has led to:

(a) characterize a single minimal and Θ-consistent explanation for a propositional abduction problem without referring to the set of all the explanations; prove the soundness and completeness of such a characterization;

(b) define a method for performing abduction in full first order logic, that does not require any preliminary transformation of formulae into normal forms; prove the soundness and completeness of such a method;

(c) single out as a central point the necessity of a relaxation of the minimality requirement in first order abduction; minimality should be defined in terms of a reasonable ordering relation $\sqsubseteq$ between C-formulae, that is decidable and such that for any α and β, if $\alpha \sqsubseteq \beta$ then $\beta \models \alpha$.

Much work still remains to be done. Here follow some of the main points to be addressed, excluding the question of minimality.

(1) Proving interesting properties of first order trees, such as "expanding an U-tree further on does not loose information" and "if two trees differ for the order of application of two rules, then any of them can be further expanded so that possibly lost information is recovered". As a consequence of these properties, the different trees that have to be generated to perform first order abduction are all instances of finite subtrees of the same tree.

(2) Formulating the first order free variable calculi on the style of [20], with the aim of reducing the dependencies of skolem terms from dummies as much as possible. In fact, a list of "dominating" dummies can be attached to every formula, that augments with every application of a γ-rule to that formula and determines the set of variables that have to occur in a skolem

term introduced by an application of a δ-rule. This mechanism reduces the failures in attempts to unify pairs of formulae, without loosing the soundness of the procedure, therefore improving efficiency.

(3) Characterizing and comparing the set of explanations that is generated by different methods (the resolution based ones and the tableau/sequent method), *before* filtering them w.r.t. minimality and consistency with the theory. If different methods do not produce the same unfiltered set, the set generated by a given method can be a measure of its efficiency.

(4) Studying abduction in modal, intuitionistic and linear logics, by means of the same proof-theoretical tools.

References

[1] A. Avron. Gentzen-type systems, resolution and tableaux. *Journal of Automated Reasoning*, 10:265–281, 1993.

[2] K. A. Bowen. Programming with full first-order logic. In J. E. Hayes, D. Michie, and Y.-H. Pao, editors, *Machine Intelligence*, volume 10, pages 421–440. Halsted Press, 1982.

[3] M. Cialdea Mayer, F. Pirri, and C. Pizzuti. Natural properties of abductive hypotheses in three valued logic. In *International Conference on Logic Programming (ICLP '93) - Postconference Workshop on Abduction, Budapest, Hungary*, 1993.

[4] P. T. Cox and T. Pietrzykowski. A complete nonredundant algorithm for reversed skolemization. *Theoretical Computer Science*, 28:239–261, 1984.

[5] P. T. Cox and T. Pietrzykowski. Causes for events: their computation and applications. In *Proceedings of the Eighth Conference on Automated Deduction (CADE-86)*, pages 608–621, 1986.

[6] M. Fitting. *First Order Logic and Automated Theorem Proving*. Springer-Verlag, New York, 1989.

[7] J. H. Gallier. *Logic and Computer Science - Foundations of Automatic Theorem Proving*. Harper & Row, New York, 1986.

[8] G. Gentzen. Untersuchungen uber das logische schliessen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935.

[9] K. Inoue. Linear resolution for consequence finding. *Artificial Intelligence*, 56:301–353, 1992.

[10] S. Kanger. A simplified proof method for elementary logic. In P. Braffort and D. Hirshberg, editors, *Computer Programming and Formal Systems*, pages 87–94. North Holland, 1963.

[11] S.C. Kleene. *Mathematical Logic*. John Wiley & Sons, New York, 1966.

[12] P. Marquis. Extending abduction from propositional to first-order logic. In *Proceedings of the First International Workshop on Fundamentals of Artificial Intelligence Research (FAIR '91)*, pages 141–155, 1991. Springer-Verlag, LNAI 535.

[13] C. G. Morgan. Hypothesis generation by machine. *Artificial Intelligence*, 2:179–187, 1971.

[14] D. Poole. Explanation and prediction: an architecture for default and abductive reasoning. *Computational Intelligence*, 5:97–110, 1989.

[15] D. Poole. Compiling a default reasoning system into prolog. *New Generation Computing*, 9:3–38, 1991.

[16] D. Poole, R. Goebel, and R. Aleliunas. Theorist: a logical reasoning system for default and diagnosis. In N. Cercone and G. Mc Calla, editors, *The Knowledge Frontier*, pages 331–352. Springer Verlag, 1987.

[17] H. E. Jr. Pople. On the mechanization of abductive logic. In *Proceedings of the Third International Joint Conference on Artificial Intelligence (IJCAI-73)*, pages 147–152, 1973.

[18] R. Reiter and J. de Kleer. Foundations of assumption-based truth maintenance systems: Preliminary report. In *Proceedings of the National Conference on Artificial Intelligence*, pages 183–188, 1987.

[19] M. Schmidt-Schauss. Implication of clauses is undecidable. *Theoretical Computer Science*, 59:287–296, 1988.

[20] N. Shankar. Proof search in intuitionistic sequent calculus. In *Proceedings of the Tenth Conference on Automated Deduction (CADE-90)*, pages 522–536, 1992.

[21] R.M. Smullyan. *First-Order Logic*. Springer-Verlag, Berlin, 1968.