Computación y Sistemas Vol. 2 No. 1 pp. 5-13
© 1998, CIC - IPN. *ISSN 1405-5546* Impreso en México

Computing Abduction in Semantic Tableaux

Atocha Aliseda
Universidad Nacional Autónoma de México
Instituto de Investigaciones Filosóficas
Circuito Mario de la Cueva s/n
Ciudad Universitaria.
México, D.F. 04510
atocha@filosoficas.unam.mx

Article received on July 28, 1998; accepted on September 14, 1998

Abstract

The aim of this paper is to propose a way to extend the framework of semantic tableaux in order to compute several forms of abductive explanations: atomic, conjunctive and in disjunctive form. Our focus is on computing consistent explanations. That is, formulas which satisfy the entailment and the consistency condition of the standard logical formulation of abduction.

Our strategy for computing abductive explanations avoids the construction of inconsistencies and it is done in a compositional fashion; abductive explanations with complex forms are constructed from simpler ones. This allows us to identify special cases, for example that in which there are no consistent atomic explanations whatsoever.

Keywords:
Abduction, Abductive Programming, Semantic Tableaux

1 Introduction

In the setting of abduction in artificial intelligence, logical as well as computational approaches are found. While the former focuses on giving a semantics to the logic of abduction, usually defined as 'backwards deduction plus additional conditions', the latter is concerned with providing algorithms to produce abductions. Ideally, computed abductions satisfy the semantic conditions of the logical formulation, and viceversa: formulas which satisfy the semantic conditions can be generated by algorithmic means.

The standard version of logic-based abduction as deduction via some consistent additional assumption, satisfying certain extra conditions is the following (it combines some common requirements from the literature: Konolige, 1990., Kakas *et al.*, 1993., Mayer and Pirri, 1993):

> Given a theory Θ (a set of formulae) and a formula φ (an atomic formula), α is an explanation if
>
> 1. $\Theta \cup \alpha \models \varphi$
> 2. α is consistent with Θ
> 3. α is 'minimal' (there are several ways to characterize minimality).
> 4. α has a syntactic restriction (usually an atomic formula).

An additional condition not always made explicit is that $\Theta \not\models \varphi$. Sometimes, the latter condition figures as a precondition for an *abductive problem.*

There are several options for treating abduction from a procedural perspective. One is standard proof analysis, as in logical proof theory. Another approach would program purely computational algorithms to produce the various types of abduction that we want.

An intermediate possibility is logic programming, which combines proof theory with an algorithmic flavor. This is a popular framework, and it has opened a whole field of abductive logic programming (ALP) (Kakas *et al.*, 1993). Another framework in line with logic programming is semantic tableaux. This is a well-motivated standard logical framework. But over these structures, different search strategies can compute several versions of abduction. Moreover, we can naturally compute various kinds of abducibles: atoms, conjunctions or even conditionals. This goes beyond the framework of abductive logic programming, in which abducibles are atoms from a special set of abducibles.

However, although there is some work on diagnosis applications using tableaux (Baumgartner *et al.*, 1997), and on implementing inductive principles for tableaux (Baaz *et al.*, 1997), there is practically no work in modeling abduction in the framework of semantic tableaux. One exception is found in (Gervás, 1995), in which tableaux are used to model natural language presuppositions, which is viewed as an abductive phenomenon. The only other exception of which we know of is found in (Mayer and Pirri, 1993), which has been a source of inspiration to our work, but has gone in different directions, both concerning strategy and output.

Modeling abduction in semantic tableaux shows that this mathematical framework has further uses which go beyond theorem proving and model checking. In our proposal for abduction, tableaux serve as structures to represent theories, and these may be extended or retracted via special strategies to compute consistent explanations.

Overview

The paper is divided into three parts. In section 2, we give a brief overview of the framework of semantic tableau and put forward a characterization of extension types. In section 3, we propose abduction as a special type of tableau extension and discuss two strategies for its computation. In section 4, we put forward algorithms to compute the several forms of abductive explanations and provide several examples. In particular, explanations with complex forms are constructed from simpler ones. This allows us to identify special cases, for example that in which there are no consistent atomic explanations whatsoever.

2 Semantic Tableaux

2.1 The Framework of Semantic Tableaux

The logical framework of semantic tableaux is a refutation method introduced in the 50's independently by Beth (Beth, 1969) and Hintikka (Hintikka, 1955). A more modern version is found in (Smullyan, 1968) and it is the one used here. The main idea of this framework is the construction of a tableau (a tree-like structure) for a theory Θ (a finite set of formulae) and a formula $\neg\varphi$ in order to prove whether $\Theta \models \varphi$. If the resulting tableau is closed (no branches are open) then Θ and $\neg\varphi$ are unsatisfiable, showing that $\Theta \models \varphi$. Otherwise, one or more counterexamples have been constructed (showed by the open branches of the tableau) and $\Theta \not\models \varphi$ is concluded.

Semantic tableaux are a *sound and complete* system:

$\Theta \models \varphi$ iff there is a closed tableau for $\Theta \cup \{\neg\varphi\}$.

And they constitute a decision method for propositional logic. This is different with predicate logic, where quantifier rules may lead to unbounded repetitions. In the latter case, the tableau method is only semi-decidable. (If the initial set of formulas is unsatisfiable, the tableau will close in finitely many steps. But if it is satisfiable, the tableau may become infinite, without terminating, recording an infinite model.) In this paper, we shall only consider the propositional case.

For convenience in what follows, we give a quick reference list of some major notions concerning tableaus.

Closed Branch : A branch of a tableau is closed if it contains some formula and its negation.

Atomically Closed Branch : A branch is atomically closed if it is closed by an atomic formula or a negation thereof.

Open branch : A branch of a tableau is open if it is not closed.

Complete branch : A branch B of a tableau is complete if for every formula in conjunctive form which occurs in B, both its conjuncts occur in B, and for every formula in disjunctive form, at least one of its disjuncts occurs in B.

Completed Tableau : A tableau $\mathcal{T}$ is completed if every branch of $\mathcal{T}$ is either closed or complete.

Proof of $\Theta \models \varphi$: A proof of $\Theta \models \varphi$ is a closed tableau for $\Theta \cup \{\neg\varphi\}$.

Tableaux are widely used in logic, and they have many further interesting properties. These can be established by simple analysis of the rules and their motivation, as providing an exhaustive search for a counter-example. This presentation incurs no loss of generality. Given a

completed tableau for a theory Θ, which we denote here as $\mathcal{T}(\Theta)$:

- If $\mathcal{T}(\Theta)$ has open branches, Θ is consistent. Each open branch corresponds to a verifying model.
- If $\mathcal{T}(\Theta)$ has all branches closed, Θ is inconsistent.

Another, more computational feature is that, given some initial verification problem, the order of rule application in a tableau tree does not affect the result. The structure of the tree may be different, but the outcome as to consistency is the same.

2.2 Characterization of Tableaux Extensions

A tableau is extended with a formula via the usual expansion rules. An extension may modify a tableau in several ways. These depend both on the form of the formula to be added and on the other formulas in the theory represented in the original tableau. If an atomic formula is added, the extended tableau is just like the original with this formula appended at the bottom of its open branches. If the formula has a more complex form, the extended tableau may look quite different (e.g., disjunctions cause every open branch to split into two). In total, however, when expanding a tableau with a formula, the effect on the open branches can only be of three types. Either (i) the added formula closes no open branch or (ii) it closes all open branches, or (iii) it may close some open branches while leaving others open. In order to compute consistent abductions, we need to clearly distinguish these three ways of extending a tableau. We label them as *open*, *closed*, and *semi-closed* extensions, respectively. In what follows we define these notions more precisely.

A propositional language is assumed with the usual connectives, whose formulas are of three types: literals (atoms or their negations), conjunctions (α-type from now on), or disjunctions (β-type).

Definition

Given a theory Θ, its corresponding *completed tableau* $\mathcal{T}(\Theta)$ is represented as the union of its branches. Each branch is the set of formulas which label that branch.

$\mathcal{T}(\Theta) = \Gamma_1 \cup \ldots \cup \Gamma_k$ where each Γ_i may be open or closed.

Definition

Given $\mathcal{T}(\Theta)$ the addition of a formula γ to each of its branches Γ_i is defined by the following $+$ operation:

- Γ_i closed: $\Gamma_i + \gamma = \Gamma_i$
- Γ_i completed open branch:

  Case 1 γ is a literal $\Gamma_i + \gamma = \Gamma_i \cup \{\gamma\}$

  Case 2 γ is an α-type ($\gamma = \alpha_1 \wedge \alpha_2$). $\Gamma_i + \gamma = ((\Gamma_i \cup \{\gamma\}) + \alpha_1) + \alpha_2$

  Case 3 γ is a β-type ($\gamma = \beta_1 \vee \beta_2$). $\Gamma_i + \gamma = \{(\Gamma_i \cup \{\gamma\}) + \beta_1), ((\Gamma_i \cup \{\gamma\}) + \beta_2\}$

That is, the addition of a formula γ to a branch is either Γ itself when it is closed or it is the union of its resulting branches. The operation $+$ is defined over branches, but it easily generalizes to tableaux as follows:

- Tableau Extension:

  $\mathcal{T}(\Theta) + \{\gamma\} =_{def} \bigcup\{\Gamma_i + \gamma \mid \Gamma_i \in \mathcal{T}(\Theta)\}$

Our notation allows also for embeddings $(\mathcal{T}(\Theta) + \{\gamma\}) + \{\beta\}$. Note that operation $+$ is just another way of expressing the usual tableau expansion rules. Therefore, each tableau may be viewed as the result of a suitable series of $+$ extension steps, starting from the empty tableau.

Definition. Branch Extension Types

Given an open branch Γ and a formula γ, the following are the possibilities to extend it:

- Open Extension:

  $\Gamma + \gamma = \delta_1 \cup \ldots \cup \delta_n$ is open if each δ_i is open.

- Closed Extension:

  $\Gamma + \gamma = \delta_1 \cup \ldots \cup \delta_n$ is closed iff each δ_i is closed.

- Semi-Closed Extension:

  $\Gamma + \gamma = \delta_1 \cup \ldots \cup \delta_n$ is semi-closed iff at least one δ_i is open and at least one δ_j is closed, for $i \neq j$.

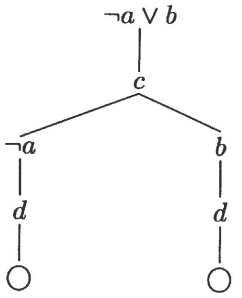
Extensions can also be defined over whole tableaux by generalizing the above definitions. A few examples will illustrate the different situations that may occur.

Examples

Let $\Theta = \{\neg a \vee b, c\}$.
$\mathcal{T}(\Theta) = [\neg a \vee b, c, \neg a] \cup [\neg a \vee b, c, b]$

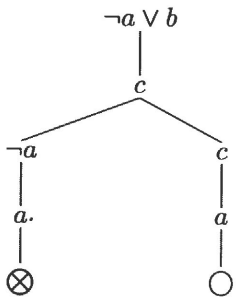
- Open Extension: $\mathcal{T}(\Theta) + \{d\}$ (d closes no branch).

  $\mathcal{T}(\Theta) + d = [\neg a \vee b, c, \neg a, d] \cup [\neg a \vee b, c, b, d]$



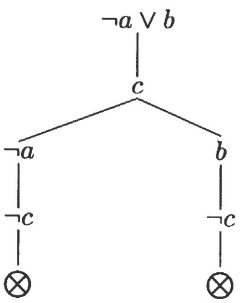
- Semi-Closed Extension: $\mathcal{T}(\Theta) + \{a\}$ (a closes only one branch).

  $\mathcal{T}(\Theta) + a = [\neg a \vee b, c, \neg a, a] \cup [\neg a \vee b, c, b, a]$



- Closed Extension: $\mathcal{T}(\Theta) + \{\neg c\}$ ($\neg c$ closes all branches)

  $\mathcal{T}(\Theta) + \neg c = [\neg a \vee b, c, \neg a, \neg c] \cup [\neg a \vee b, c, b, \neg c]$



Finally, to recapitulate an earlier point, these types of extension are related to consistency in the following way:

- Consistent Extension:

  If $\mathcal{T}(\Theta) + \{\gamma\}$ is open or semi-closed, then $\mathcal{T}(\Theta) + \{\gamma\}$ is a consistent extension.

- Inconsistent Extension:

  If $\mathcal{T}(\Theta) + \{\gamma\}$ is closed, then $\mathcal{T}(\Theta) + \{\gamma\}$ is an inconsistent extension.

Given this characterization, it is easy to calculate for a given tableau, the sets of literals for each type of extension. In our example above these sets are as follows:

$$\text{Open} = \{x \mid x \neq a, x \neq \neg c, x \neq \neg b\}$$

$$\text{Semi-closed} = \{a, \neg b\}$$

$$\text{Closed} = \{\neg c\}$$

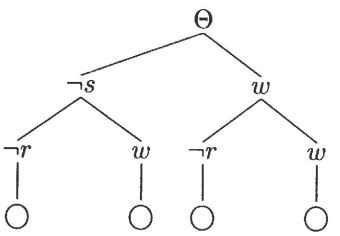
These constructions will be very useful for the calculations of abductions.

3 Abduction in Tableaux: The Main Ideas

In this section we will show the main idea for performing abduction as a kind of tableau extension. To simplify the notation from now on, we write $\Theta + \neg\varphi$ for $\mathcal{T}(\Theta) + \{\neg\varphi\}$. Moreover, in the graphical representation of a tableau, we omit the original formulas in the tableau, showing only the subformulas representing them.

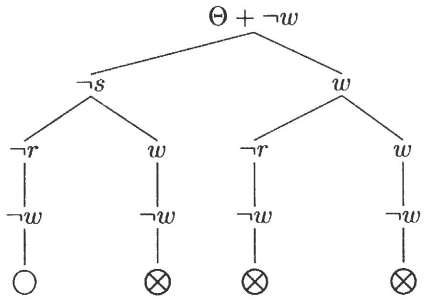
Example

Let $\Theta = \{s \rightarrow w, r \rightarrow w\}$, and let $\varphi = w$.
A tableau for Θ is as follows:



The result is an open tableau. Therefore, the theory is consistent and each open branch corresponds to a verifying model. For example, the second branch (from left to right) indicates that a model for Θ is given by making s false and w true, so we get two possible models out of this branch (one in which r is true, the other in which it is false). Generally speaking, when constructing the tableau, the possible valuations for the formulas are depicted by the branches (either $\neg s$ or w makes the first split, then for each of these either $\neg r$ or w).

When formulas are added (thereby extending the tableau), some of these possible models may disappear, as branches start closing. For instance, when $\neg\varphi$ is added (i.e. $\neg w$), the result is the following tableau for $\Theta + \neg\varphi$:



Notice that, although the resulting theory remains consistent, all but one branch has closed. In particular, most models we had before are no longer here, as w is no longer true. There is still an open branch, indicating there is a model satisfying $\Theta \cup \neg w$ (s, r, w false), which indicates that $\Theta \not\models w$.

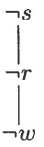
3.1 The Main Ideas

An attractive feature of the tableau method is that when φ is not a valid consequence of Θ, we get all cases in which the consequence fails graphically represented by the open branches (as shown above, the latter may be viewed as descriptions of models for $\Theta \cup \neg\varphi$.)

This fact suggests that if these *counterexamples* were 'corrected by amending the theory', through adding more premises, we could perhaps make φ a valid consequence of some (minimally) extended theory Θ'. This is indeed the whole issue of abduction. Accordingly, abduction may be formulated in this framework as a process of *expansion*, extending a tableau with suitable formulas that close the open branches.

Our example above is indeed the paradigmatic example of abduction in artificial intelligence. The theory (Θ) is composed by the rules: the lawn is wet if the sprinklers are on ($s \rightarrow w$), and the lawn is wet if it rains ($r \rightarrow w$). The fact to be explained (φ) is the lawn is wet (w). Therefore, the tableau is constructed with $\Theta + \neg w$ and the goal is to extend it in such a way that an added formula closes the open branches. That is, with an explanation for the observed fact.

In this example, the remaining open branch had the following relevant (literal) part:



The following are (some) formulas whose addition to the tableau would close this branch (and hence, the whole tableau):

$$\{s, r, w, s \wedge r, s \wedge w, r \wedge w, s \wedge \neg w, s \vee r\}$$

Note that several forms of statement may count here as abductions. In particular, those in disjunctive form (e.g. the spinklers are on or it is raining: $s \vee r$) create two branches, which then both close. (We will consider these various cases in detail later on.). Moreover, notice that all but one formula of this set (the lawn is wet: w, the trivial solution) is a semi-closed extension of the original tableau for Θ.

3.2 Generating Abductions: Two Strategies

In principle, we can compute abductions in accord with the logical formulation (cf. section 1). A direct way of doing so is as follows:

> First compute abductions according to the entailment condition (1) and then eliminate all those which do not comply with the consistency condition (2).

This strategy first translates our abductive formulations to the setting of semantic tableaux (using our extension characterization) as follows:

> Given Θ (a set of formulae) and φ (a sentence), α is an abductive explanation if:
>
> 1. $((\Theta + \neg\varphi) + \alpha)$ is a closed extension. $(\Theta, \alpha \models \varphi)$.
> 2. $(\Theta + \alpha)$ is an open extension $(\Theta \not\models \neg\alpha)$

In addition to the 'abductive conditions' we must state constraints over our search space for abducibles, as the set of formulas fulfilling any of the above conditions is in principle infinite. Therefore, we impose restrictions on the vocabulary as well as on the form of the abduced formulas:

- Restriction on Vocabulary

  α is in the vocabulary of the theory and the observation:

  $\mathrm{Voc}(\alpha) \subseteq \mathrm{Voc}(\Theta \cup \{\varphi\})$, where $\mathrm{Voc}(\psi)$ is the set of atomic formulae occurring in ψ.

- Restriction on Form

  The syntactic form of α is either a literal, a conjunction of literals (without repeated conjuncts), or a disjunction of literals (without repeated disjuncts).

Once it is clear what our search space for abducibles is, we continue with our discussion. Note that the computation of *consistent abductions* involves inspection of both closed and open branches. The algorithm would proceed in the following two steps:

- *Generating Consistent Abductions* (First Version)

  1. Generate all formulas α for which $((\Theta + \neg\varphi) + \alpha)$ is a closed extension.
  2. Select all those α from above for which $(\Theta + \alpha)$ is a closed extension.

In particular, an algorithm producing consistent abductions along these lines must produce all explanations that are inconsistent with Θ. This means many ways of closing $\mathcal{T}(\Theta)$, which will then have to be removed in Step 2. This is of course wasteful. Even worse, when there are no consistent explanations (besides the trivial one), so that we would want to give up, our procedure still produces the inconsistent ones.

Of course, there is a preference for procedures that generate abductions in a reasonably efficient way. We will show how to devise these, making use of the representation structure of tableaux, in a way which avoids the production of inconsistent formulae. Here is our idea.

- *Generating Consistent Abductions* (Second Version)

  1. Generate all formulas α for which $(\Theta + \alpha)$ is a semi-closed extension.
  2. Select those formulas α produced above for which $((\Theta + \neg\varphi) + \alpha)$ is a closed extension.

That is, first produce formulas which extend the tableau for the background theory in a consistent way (but closing at least one branch), and then check which of these are abductive explanations. As we will show later, the consistent formulae produced by the second procedure are not necessarily wasteful. They might be 'partial explanations' (an ingredient for explanations in conjunctive form), or part of explanations in disjunctive form. Moreover, it turns out that in the atomic and conjunctive cases explanations are sometimes necessarily inconsistent, therefore we identify these cases and prevent our algorithm from doing anything at all (so that we do not produce formulae which are discarded afterwards).

4 Computing Abductions

Our strategy for computing consistent abductions in semantic tableaux will be as follows. We will be using tableaux as an ordinary consequence test, while being careful about the search space for potential abducibles. The computation is divided into different forms of explanations. Atomic explanations come first, followed by conjunctions of literals, to end with those in disjunctive form. Here we sketch the algorithms for constructing each form of explanation.

Our algorithm follows version 2 of the strategies sketched earlier. That is, it first constructs those semi-closed extensions on the original tableau for Θ (which are all consistent) and then selects which of these is a closed extension for $\Theta + \neg\varphi$. The resulting α are considered the output explanations. This way we avoid the production of any inconsistency whatsoever.

4.1 Atomic Explanations

When computing consistent atomic explanations, we want to avoid any computation when there are only inconsistent atomic explanations (besides the trivial one). Here is an observation which helps us get one major problem out of the way. Atomic explanations are necessarily inconsistent when $\Theta + \neg\varphi$ is an open extension. So, we can prevent our algorithm from producing anything at all in this case.

Fact 1 *Whenever $\Theta + \neg\varphi$ is an open extension, and α a non-trivial atomic abductive explanation (different from φ), it follows that Θ, α is inconsistent.*

Proof.

Let $\Theta + \neg\varphi$ be an open extension and α an atomic explanation ($\alpha \neq \varphi$). The latter implies that $((\Theta + \neg\varphi) + \alpha)$ is a closed extension. Therefore, $\Theta + \alpha$ must be a closed extension, too, since φ closes no branches. But then, $\Theta + \alpha$ is an inconsistent extension. I.e. Θ, α is inconsistent. $\dashv$

This result cannot be generalized to more complex forms of abducibles. (We will see later that for explanations in disjunctive form, open extensions need not lead to inconsistency.) In case $\Theta + \neg\varphi$ is a semi-closed extension, we have to do real work, however, and follow the strategy sketched above. The key point in the algorithm is this. Instead of building the tableau for $\Theta + \varphi$ directly, and working with its open branches, we must start with the open branches of Θ.

1. $\Theta, \neg\varphi$ are given as input and are such that: $\Theta \not\models \varphi$, $\Theta \not\models \neg\varphi$.
2. Calculate $\Theta + \neg\varphi$. If it is an open extension, then there are no atomic consistent explanations (by fact 1), go to Step 6.

3. Calculate the set of literals $\{\gamma_1, \ldots, \gamma_n\}$ for which $\Theta + \gamma_i$ is a semi-closed extension.
4. Select from above set those γ_i for which $(\Theta + \neg\varphi) + \gamma_i$ is a closed extension.
5. The set above is the set of Consistent Atomic Explanations.
6. END.

4.2 Conjunctive Explanations

Single atomic consistent explanations may not always exist, or they may not be the only ones of interest. The case of explanations in conjunctive form ($\alpha = \alpha_1 \wedge \ldots \wedge \alpha_n$) is similar to the construction of atomic explanations. We look for literals that close branches, but in this case we want to get the literals that close some but not all of the open branches. These are the conjuncts of a 'conjunctive explanation'. Each of these conjuncts α_i may be considered as a *partial explanation*, that which contributes to part of the explaining.

As a consequence of this characterization, no partial explanation is an atomic explanation. That is, a conjunctive explanation must be a conjunction of partial explanations. The motivation is this. We want to construct explanations which are *non-redundant*, in which every literal does some explaining. Moreover, this condition allows us to bound the production of explanations in our algorithm. We do not want to create what are intuitively 'redundant' combinations. For example, if p and q are abductive explanations, then $p \wedge q$ should not be produced as explanation.

For conjunctive explanations, we can also avoid any computation when there are only 'blatant inconsistencies', essentially by the same observation as before.

Fact 2 *Whenever $\Theta + \neg\varphi$ is an open extension, and $\alpha = \alpha_1 \wedge \ldots \wedge \alpha_n$ is a conjunctive abductive explanation, it holds that Θ, α is inconsistent.*

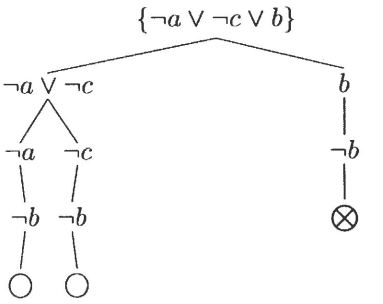
The proof is analogous to that for the atomic case.

In order to construct partial explanations, the ingredients of conjunctive explanations, the key idea is to build conjunctions out of those formulas for which both $\Theta + \gamma_i$ and $(\Theta + \neg\varphi) + \gamma_i$ are semiclosed extensions. The reason is as follows: the former condition assures that these formulas do close at least one branch from the original tableau. The latter condition discards atomic explanations (for which $(\Theta + \neg\varphi) + \gamma_i$ is closed) and the trivial solution ($\gamma_i = \varphi$ when $(\Theta + \neg\varphi) + \gamma_i$ is open). Conjunctions are then constructed out of these formulas and those which induce a closed extension for $\Theta + \neg\varphi$ are the selected conjunctive explanations.

Here is the algorithm sketch:

1. $\Theta, \neg\varphi$ are given as input and are such that: $\Theta \not\models \varphi$, $\Theta \not\models \neg\varphi$.
2. Calculate $\Theta + \neg\varphi$. If it is an open extension, then there are no conjunctive consistent explanations (by fact 1), go to Step 8.
3. Calculate the set of literals $\{\gamma_1, \ldots, \gamma_n\}$ for which $\Theta + \gamma_i$ is a semi-closed extension.
4. Select from above set those γ_i for which $(\Theta + \neg\varphi) + \gamma_i$ is a semi-closed extension. This is the set of Partial Explanations: $\{\gamma_1, \ldots, \gamma_m\}$, $m \leq n$.
5. Construct conjunctions from set of partial explanations starting in length k (number of open branches) to end in lenght m (number of partial explanations). Label these conjunctions as follows: $\rho_1, \ldots, \rho_s$.
6. Select those ρ_i from above for which $(\Theta + \neg\varphi) + \rho_i$ is a closed extension.
7. The set above is the set of Consistent Conjunctive Explanations.
8. END.

Here is an example:
Let $\Theta = \{\neg a \vee \neg c \vee b\}$, and $\varphi = b$. The corresponding tableau is as follows:



Following the algorithm above, we have that the set of literals $\{\gamma_1, \ldots, \gamma_n\}$ for which $\Theta + \gamma_i$ is a semi-closed extension is $\{a, c, \neg b\}$, and the set of those for which $(\Theta + \neg\varphi) + \gamma_i$ is a semi-closed extension is $\{a, c\}$. Thus, $a \wedge b$ is the one and only explanation in conjunctive form.

4.3 Disjunctive Explanations

As for disjunctive explanations, unfortunately, we no longer have the clear cut distinction between open and semi-closed extensions to know when there are only inconsistent explanations. The reason is that for explanations α in disjunctive form, $(\Theta + \neg\varphi)$ open and $((\Theta + \neg\varphi) + \alpha)$ closed does not imply that $\Theta + \alpha$ is closed because α generates two branches.

The key issue in the construction of the algorithm to compute disjunctive consistent explanations is the following:

1. Construct disjunctive formulas by combining the atomic and conjunctive consistent ones above: $\{\gamma_1, \ldots, \gamma_n\}$
2. For each formula X in Θ, construct a disjunction as follows: $\neg X \vee \varphi$.

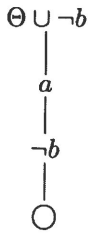
All disjunctions constructed above are consistent with Θ. Moreover, they are such that added to $(\Theta + \neg\varphi)$ produce a closed extension. Each disjunct of item 1 is an abductive explanation, therefore the disjunction is. It is also easy to see that formulas constructed as in item 2 are consistent abductive explanations.

The construction as in item 2 suggests is that there are always consistent explanations in disjunctive form, provided that the theory is consistent:

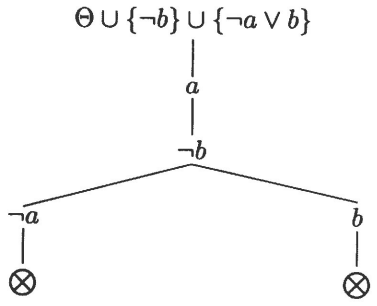
Fact 3 *Given that $\Theta + \neg\varphi$ is a consistent extension, there exists an abductive consistent explanation in disjunctive form.*

The key point to prove this fact is that an explanation may be constructed as in item 2 above.

Here is an example:
Let $\Theta = \{a\}$ $\varphi = b$.
The tableau structure for $\mathcal{T}(\Theta \cup \neg b)$ is as follows:



The only atomic consistent explanation is b, the trivial one. Moreover, there are no 'partial explanations' as there is only one open branch. So, there are no explanations in conjuctive form. Therefore, step 1 of the above algorithm does not apply. As for step 2, since the only formula in Θ is a we construct the disjunction $\neg a \vee b$, which is the consistent abductive explanation in disjunctive form. The effect of adding it to the tableau is as follows:



This examples serves as a representation of a case in which a causal connection is found between a formula in the theory a (e.g. it is raining), and a fact to be explained (b) (e.g. the lawn is wet), namely that rain causes de lawn to be wet ($a \rightarrow b$).

5 Conclusions

This paper proposes a way to compute abduction with semantic tableau. For this purpose an appropriate extension of this framework was put forward characterizing several tableau extensions. Over this framework, several algorithmic constructions were sketched to compute different forms of explanations: atomic, conjunctive, and in disjunctive form. Each of these constructions may constitute a module in a program for computing abductions (for the complete program, see (Aliseda, 1997). But this separation already gives us some account of *minimal explanations*, when minimality is regarded as simplicity. As for conjunctive explanations, as we have noted before, their construction is one of computing 'partial explanations', those which partially explain a fact by closing some, though not all open branches for its refutation. Finally, explanations in disjunctive form can be constructed in various ways. E.g., one can combine atomic explanations, conjunctive explanations or form a conditional with any formula of the theory Θ and φ. This reflects our view that explanations are built in a compositionl fashion: complex solutions are constructed from simpler ones.

Semantic tableaux are a natural vehicle for implementing abduction. They allow for a clear formulation of what counts as an abductive explanation, while being flexible and suggestive as to possible modifications and extensions. Derivability and consistency, the ingredients of consistent abductions, are indeed a natural blend in tableaux, because we can manipulate open and closed branches with equal ease.

Even so, our actual algorithms were more than a straight transcription of the logical formulation (which might be very inefficient). Our computational strategy provided algorithmic constructions which produce consistent formulas, selecting those which count as explanations. The specification of each of these constructions as well as the constraints on the vocabulary and form of explanations, allowed to restrict the space search for abducibles, in order to produce a finite number of explanations which are non-redundant and non-trivial.

Our proposal has several advantages over existing others. In particular, in (Mayer and Pirri, 1993) propositional algorithms over semantic tableau produce minimal atomic explantions or nothing at all. Moreover, their implementation follows version 1 of the strategies sketched earlier, a way in which it is impossible to avoid the production of inconsistencies. However, their proposal treats abduction in first order logic, something which we left aside. Their analysis shows that while propositional abduction is easy to compute, even considering minimality, first-order abduction is inherently undecidable. This may be the reason why most logic-

based approaches to abduction are restricted to propositional or grounded theories (cf. Kakas *et al.*, 1993., Konolige, 1990).

There are still further uses, though which go beyond our analysis so far, but may be found in (Aliseda 1997). They include various logical aspects of tableau abduction, plus soundness and completness of our algorithms. Moreover, abduction as revision can also be implemented in semantic tableaux, involving a contraction algorithm to 'open branches' over tableaux. Finally, in our view, abduction is not a new notion of inference tied to a specific logic, but rather a topic-dependent practice of explanatory reasoning, which can be supported by various types of logics. This idea suggests itself as an extension to this work, namely, to take as a point of departure existing proposals which use semantic tableaux for modal and intuitionistic logics (cf. Baumgartner *et al.*, 1995., Galmiche, 1997), and extend their framework to account for abduction.

Acknowledgements

I am grateful to Johan van Benthem and Marianne Kalsbeek for many useful discussions on earlier drafts to this paper, and to two anonymous referees for their helpful comments and suggestions.

References

Aliseda, A., *Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence.* PhD dissertation. Stanford University. Published by the Institute for Logic, Language and Computation. University of Amsterdam, The Netherlands. (ILLC Dissertation Series 1997-4). 1997.

Baumgartner, P., Hahnle, R. and Posegga J. (Eds.). *Theorem Proving with Analytic Tableaux and Related Methods, 4th International Workshop, TABLEAUX '95, Proceedings.* Lecture Notes in Artificial Intelligence 918, Subseries of Lecture Notes in Computer Science. Springer Verlag 1995.

Baumgartner, P., Frohlich, P., Furbach U., and Nejdl W. "Tableaux for Diagnosis Applications", in **Galmiche, D. (Ed.).** *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX '97, Proceedings*, pages 76–90. Lecture Notes in Artificial Intelligence 1227, Subseries of Lecture Notes in Computer Science. Springer Verlag 1997.

Baaz, M., Egly, U., Fermuller, C, C.G. "Lean Induction Principles for Tableaux", in **Galmiche, D. (Ed.).** *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX '97, Proceedings*, pages 62–75. Lecture Notes in Artificial Intelligence 1227, Subseries of Lecture Notes in Computer Science. Springer Verlag 1997.

Beth, E.W.., "Semantic Entailment and Formal Derivability". In J. Hintikka (ed), *The Philosophy of Mathematics*, p. 9–41. Oxford University Press. 1969.

Galmiche, D. (Ed.). *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX '97, Proceedings.* Lecture Notes in Artificial Intelligence 1227, Subseries of Lecture Notes in Computer Science. Springer Verlag 1997.

Gervás, P. *Logical considerations in the interpretation of presuppositional sentences.* PhD Thesis, Department of Computing, Imperial College London. 1995.

Hintikka, J., "Two Papers on Symbolic Logic: Form and Quantification Theory and Reductions in the Theory of Types", *Acta Philosophica Fennica*, Fasc VIII, 1955.

Kakas, A.C., Kowalski, R.A., and Toni, F., "Abductive Logic Programming", *Journal of Logic and Computation* 2(6) (1993) 719-770.

Konolige, K., "A General Theory of Abduction". In: *Automated Abduction, Working Notes*, pp. 62–66. Spring Symposium Series of the AAA. Stanford University. 1990.

Mayer, M.C. and Pirri, F. "First order abduction via tableau and sequent calculi", in *Bulletin of the IGPL*, vol. 1, pp.99–117. 1993.

Smullyan, R.. *First Order Logic.* Springer Verlag, 1968.

Atocha Aliseda *received a PhD degree in Philosophy and Symbolic Systems from Stanford University in 1997. Her PhD thesis "Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence" is published by the Institute for Logic, Language and Information (ILLC), University of Amsterdam, 1997. She is currently associate professor at the Instituto de Investigaciones Filosóficas at the Universidad Nacional Autónoma de México in Mexico City. Her main research interests are: non-monotonic logics, knowledge representation, knowledge discovery and the connection between philosophy of science and artificial intelligence.*