A first approach to Abductive Defeasible Logic Programming: formalization and properties

Mauro J. Gómez Lucero Alejandro J. García Carlos I. Chesñevar Guillermo R. Simari

Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Laboratorio de Investigación y Desarrollo de Inteligencia Artificial (LIDIA)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Bahía Blanca, B8000CPB, Argentina
{mjg, ajg, grs, cic}@cs.uns.edu.ar

Abstract

This article presents the confluence of two general ideas: Defeasible Logic Programming (DeLP, an argumentation based formalism for representing knowledge and reasoning) and Abduction in logic. In first place, we introduce a framework that formally states the problem of doing abduction (obtaining abductive explanations) in the particular case that the abductive theory is a DeLP Program. In the reminder of this work we face that problem, providing a formal characterization of the notion of abductive explanation, in such a way that we could then easily calculate the explanations from this formal characterization. An important advantage of this approach is that by proving the correctness of the characterization, we ensure the correctness of the method for obtaining explanations.

Keywords: Defeasible Reasoning, Abduction, Knowledge Representation and Reasoning.

1 INTRODUCTION

In this work we will present the confluence of two general ideas. Firstly, Defeasible Logic Programming (DeLP) is a formalism for representing knowledge and reasoning, that combines results of Logic Programming and Defeasible Argumentation. This formalism will be briefly introduced in section 2. Secondly, Abduction in logic represents a form of reasoning where, given a (logic) theory T and a sentence G, we try to find a set of sentences Φ (abductive explanation for G) such that $T \cup \Phi \vdash G$ and $T \cup \Phi$ is consistent.

Here, we face the problem of doing abduction in the particular case that the abductive theory is a (restricted) DeLP program $\mathcal{P}$, and the sentence to be explained is a literal in the language of $\mathcal{P}$. In order to formally state the problem, in section 3 we introduce the notion of Abductive-DeLP framework, and we present a notion of abductive explanation slightly adapted for that framework.

In particular, we will focus on the problem of obtaining abductive explanations (for a literal h in an Abductive-DeLP framework AF). The strategy adopted consists in providing a formal characterization of the notion of abductive explanation, in such a way that we could then easily calculate the explanations from this formal characterization. An important advantage of this approach is that by proving the correctness of the characterization (not done in this work for space reasons), we ensure the correctness of the method for obtaining explanations.

The formal characterization of the notion of explanation involves two parts. From the definition of abductive explanation, and considering the reasoning mechanism of DeLP, we know that the incorporation of an abductive explanation for a given literal to the abductive theory (DeLP program), causes

Partially financed by CONICET (PIP 5050), Universidad Nacional del Sur (PGI 24/ZN11) and Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096).

the *emergence* or *activation* of new arguments (and possibly, the deactivation of existent ones), which in turn causes the literal to be accepted. Then in section 4 we formalize which information should be present in an explanation (and which should not) in order to activate a given argument. On the other hand, in section 5 we formally state which arguments must the explanation activate and which must not in order to effectively explain the acceptance of the literal. Finally, combining both, we will have a formal characterization of the notion of abductive explanation.

2 DELP OVERVIEW

Defeasible Logic Programming (DeLP) combines results of Logic Programming and Defeasible Argumentation. The system is fully implemented and is available online [1]. A brief explanation is included below (see [2]for full details). A DeLP program $\mathcal{P}$ is a set of facts, strict rules and defeasible rules. *Facts* are ground literals representing atomic information or the negation of atomic information using strong negation "$\sim$" (*e.g.,* $chicken(little)$ or $\sim scared(little)$). *Strict Rules* represent non-defeasible information and are denoted $L_0 \leftarrow L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals (*e.g.,* $bird \leftarrow chicken$ or $\sim innocent \leftarrow guilty$). *Defeasible Rules* represent tentative information and are denoted $L_0 \prec L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals (*e.g.,* $\sim flies \prec chicken$ or $flies \prec chicken, scared$).

When required, $\mathcal{P}$ is denoted (Π, Δ) distinguishing the subset Π of facts and strict rules, and the subset Δ of defeasible rules (see Ex. 2.1). *Strong negation* is allowed in the head of rules, and hence may be used to represent contradictory knowledge. From a program (Π, Δ) contradictory literals could be derived. Nevertheless, the set Π (which stands for non-defeasible information) must possess certain internal coherence, *i.e.*, no pair of contradictory literals can be derived from Π.

A defeasible rule is used to represent tentative information that may be used if nothing could be posed against it. Observe that strict and defeasible rules are ground. However, following the usual convention [3], some examples use "schematic rules" with variables.

Example 2.1. *Consider the DeLP program* $(\Pi_{2.1}, \Delta_{2.1})$ *where:*

$$\Pi_{2.1} = \left\{ \begin{array}{l} bird(X) \leftarrow chicken(X) \\ chicken(tina) \\ chicken(little) \\ scared(tina) \end{array} \right\} \quad \Delta_{2.1} = \left\{ \begin{array}{l} flies(X) \prec bird(X) \\ flies(X) \prec chicken(X), scared(X) \\ \sim flies(X) \prec chicken(X) \end{array} \right\}$$

This program has three defeasible rules representing tentative information about the flying ability of birds in general, and about regular chickens and scared ones. It also has a strict rule expressing that every chicken is a bird, and three facts: 'tina' and 'little' are chickens, and 'tina' is scared.

Given a DeLP program $\mathcal{P}$, a *defeasible derivation* for a literal L from $\mathcal{P}$ is a (tentative) proof for L involving strict and defeasible rules in the program. The reason for using the term defeasible is that it is possible to derive contradictory literals from $\mathcal{P}$. For example, from $(\Pi_{2.1}, \Delta_{2.1})$ in Ex. 2.1 it is possible to derive $flies(tina)$ and $\sim flies(tina)$.

For the treatment of contradictory knowledge DeLP incorporates a defeasible argumentation formalism. This formalism allows the identification of those pieces of knowledge that are in contradiction, and a *dialectical process* is used for deciding which information prevails as warranted. This dialectical process (see below) involves the construction and evaluation of arguments that either support or interfere with the query under analysis. As we will show next, arguments supporting the answer for a given query will be shown in a particular way using *dialectical trees*. The definition of dialectical tree will be included below, but first, we will give a brief explanation of other related concepts (for the details see [2]).

Definition 2.1 (Argument Structure). *Let* (Π, Δ) *be a DeLP program. . We will say that* $\langle \mathcal{A}, \alpha \rangle$ *is an argument structure (or just argument) for a literal* α *from* (Π, Δ)*, if* $\mathcal{A}$ *is the minimal set of defeasible rules (*$\mathcal{A} \subseteq \Delta$*), such that: (1) there exists a defeasible derivation for* α *from* $\Pi \cup \mathcal{A}$*, and (2) the set* $\Pi \cup \mathcal{A}$ *is non-contradictory.*

Example 2.2. *From the DeLP program* $(\Pi_{2.1}, \Delta_{2.1})$ *the following arguments can be obtained:*
$\langle \mathcal{A}_1, flies(tina) \rangle$, *where* $\mathcal{A}_1 = \{flies(tina) \prec bird(tina)\}$
$\langle \mathcal{A}_2, {\sim}flies(tina) \rangle$, *where* $\mathcal{A}_2 = \{{\sim}flies(tina) \prec chicken(tina)\}$
$\langle \mathcal{A}_3, flies(tina) \rangle$, *where* $\mathcal{A}_3 = \{flies(tina) \prec chicken(tina), scared(tina)\}$

A literal L is *warranted* if there exists a non-defeated argument $\mathcal{A}$ supporting L. To establish if $\langle \mathcal{A}, \alpha \rangle$ is a non-defeated argument, *defeaters* for $\langle \mathcal{A}, \alpha \rangle$ are considered, *i.e.*, counter-arguments that by some criterion are preferred to $\langle \mathcal{A}, \alpha \rangle$. In DeLP, the comparison criterion is usually *generalized specificity*, but in the examples given in this paper we will abstract away from this particular criterion, since in this work it introduces unnecessary complications. Since defeaters are arguments, there may exist defeaters for them, and defeaters for these defeaters, and so on. Thus, a sequence of arguments called *dialectical line* is constructed, where each argument defeats its predecessor. In order to avoid undesirable sequences that may represent circular or fallacious argumentation lines, in DeLP a dialectical line is *acceptable* if it satisfies certain constraints (see [2]).

Example 2.3. *From Ex. 2.2, we have that argument* $\langle \mathcal{A}_2, {\sim}flies(tina) \rangle$ *defeats* $\langle \mathcal{A}_1, flies(tina) \rangle$*, argument* $\langle \mathcal{A}_3, flies(tina) \rangle$ *is a defeater for* $\langle \mathcal{A}_2, {\sim}flies(tina) \rangle$*, and the sequence of arguments* $[\langle \mathcal{A}_1, flies(tina) \rangle$*,* $\langle \mathcal{A}_2, {\sim}flies(tina) \rangle$*,* $\langle \mathcal{A}_3, flies(tina) \rangle]$ *is an acceptable argumentation line.*

Clearly, there might be more than one defeater for a particular argument. Therefore, many acceptable argumentation lines could arise from a given argument, leading to a tree structure. Before defining the notion of *dialectical tree*, we need to introduce some terminology. Let $\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle]$ be an acceptable argumentation line, and $\langle \mathcal{B}_i, q_i \rangle$ be a defeater of $\langle \mathcal{A}_n, h_n \rangle$ such that $\Lambda' = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle, \langle \mathcal{B}_i, q_i \rangle]$ is also acceptable. Then we will say that $\langle \mathcal{B}_i, q_i \rangle$ *extends* Λ.

Definition 2.2 (Dialectical tree [2]). *Let* $\langle \mathcal{A}_0, h_0 \rangle$ *be an argument from a program* $\mathcal{P}$*. A dialectical tree for* $\langle \mathcal{A}_0, h_0 \rangle$ *from* $\mathcal{P}$*, denoted* $\mathcal{T}_{\mathcal{P}}(\langle \mathcal{A}_0, h_0 \rangle)$*, is defined as follows:*

(1) The root of the tree is labelled with $\langle \mathcal{A}_0, h_0 \rangle$*.*
(2) Let N *be a node of the tree labelled* $\langle \mathcal{A}_n, h_n \rangle$*, and* $\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle]$ *be the (acceptable) argumentation line corresponding to the sequence of labels of the path from the root to* N*. Let* { $\langle \mathcal{B}_1, q_1 \rangle$, $\langle \mathcal{B}_2, q_2 \rangle$, …, $\langle \mathcal{B}_k, q_k \rangle$ } *be the set of all arguments extending* Λ*. Then for each argument* $\langle \mathcal{B}_i, q_i \rangle$*, the node* N *has a child* N_i *labelled* $\langle \mathcal{B}_i, q_i \rangle$*. If there exist no argument extending* Λ*, then* N *is a leaf.*

In a dialectical tree, every node (except the root) represents a defeater of its parent, and leaves correspond to non-defeated arguments. Each path from the root to a leaf corresponds to a different acceptable argumentation line. A dialectical tree provides a structure for considering all the possible acceptable argumentation lines that can be generated for deciding whether an argument is ultimately defeated. We call this tree *dialectical* because it represents an exhaustive dialectical analysis for the argument in its root.

Given a literal h and an argument $\langle \mathcal{A}, h \rangle$ from a program $\mathcal{P}$, to decide whether a literal h is warranted, every node in the dialectical tree $\mathcal{T}_{\mathcal{P}}(\langle \mathcal{A}, h \rangle)$ is recursively marked as "D" (*defeated*) or "U" (*undefeated*), obtaining a marked dialectical tree $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h \rangle)$ as follows:

1. All leaves in $\mathcal{T}^*_{\mathcal{P}}(\langle\mathcal{A}, h\rangle)$ are marked as "U"s, and
2. Let N be an inner node of $\mathcal{T}^*_{\mathcal{P}}(\langle\mathcal{A}, h\rangle)$. Then N will be marked as "U" iff every child of N is marked as "D". The node N will be marked as "D" iff it has at least a child marked as "U".

Given an argument $\langle\mathcal{A}, h\rangle$ obtained from $\mathcal{P}$, if the root of $\mathcal{T}^*_{\mathcal{P}}(\langle\mathcal{A}, h\rangle)$ is marked as "U", then we will say that $\mathcal{T}^*_{\mathcal{P}}(\langle\mathcal{A}, h\rangle)$ *warrants* h and that h is *warranted* from $\mathcal{P}$ (that will be noted $\mathcal{P} \mid\!\sim_W h$).

In this paper, an argument $\langle\mathcal{A}, h\rangle$ will be depicted as a triangle, whose upper vertex is labelled with the conclusion h, and the set of defeasible rules $\mathcal{A}$ are associated with the triangle itself. Marked dialectical trees will be depicted as shown in figure 1.

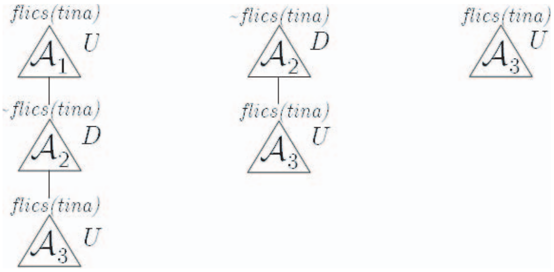


Figure 1: Dialectical trees for $flies(tina)$.

Example 2.4. *(Extends Ex. 2.3) Figure 1 shows the marked dialectical tree* $\mathcal{T}^*_{\mathcal{P}_{2.1}}(\langle\mathcal{A}_1, f\rangle)$ *(the leftmost tree), which has only one argumentation line. Observe that the argument* $\langle\mathcal{A}_2, \sim f\rangle$ *interferes with the warrant of 'flies(tina)' and the argument* $\langle\mathcal{A}_3, f\rangle$ *reinstates* $\langle\mathcal{A}_1, f\rangle$*. The root of* $\mathcal{T}^*_{\mathcal{P}_{2.1}}(\langle\mathcal{A}_1, f\rangle)$ *is marked as "U" and therefore the literal 'flies(tina)' is warranted.*

3 ABDUCTIVE DELP FRAMEWORK

In this section we will introduce a framework for doing abduction in a (restricted) version of DeLP without strict rules. Before, we need to introduce certain distinguished sets of literals associated with any set of defeasible rules. Let Δ be a set of defeasible rules. Then we will use $all\_lits(\Delta)$ to denote the set of all literals appearing in defeasible rules in Δ. Also, we will use $head\_lits(\Delta)$ to denote the set of all literals that are heads of defeasible rules in Δ (*i.e.*, $head\_lits(\Delta) = \{H | H \prec B_1, B_2, ..., B_n \in \Delta\}$). Finally, we will use $body\_lits(\Delta)$ to denote the set of all literals that appear in the body of a defeasible rule in Δ. (*i.e.*, $body\_lits(\Delta) = \{B_i | H \prec B_1, ..., B_i, ..., B_n \in \Delta\}$).

Definition 3.1 (A-DeLP Framework). *An A-DeLP framework is a pair* $\langle\Pi_f \cup \Delta, A\rangle$ *where*

- $\Pi_f \cup \Delta$*, the theory, is a DeLP program without strict rules, ie.* Π_f *is a consistent set of literals (called facts) and* Δ *is a set of defeasible rules.*
- *A is a set of distinguished literals in the language of* $\Pi_f \cup \Delta$*, called abducibles, such that*

$$body\_lits(\Delta) \backslash (heads(\Delta) \cup \Pi_f \cup \overline{\Pi_f}) \subseteq A \subseteq (all\_lits(\Delta) \cup \overline{all\_lits(\Delta)}) \backslash (\Pi_f \cup \overline{\Pi_f})$$

Given a literal Q, an abductive explanation for Q w.r.t. the abductive framework, is defined as a set of abducibles Φ ($\Phi \subseteq A$) such that:

- $\Pi_f \cup \Delta \cup \Phi \mid\!\sim_W Q$ and
- $\Pi_f \cup \Phi$ is consistent.

The set A associated with an A-DeLP framework constitutes the domain of the explanations in that framework, *i.e.*, A is the set of literals (abducibles) that can integrate the explanations. The set A must be specified when defining the A-DeLP framework and it could be any set of literals satisfying the restrictions stated by the previous definition. The election of a specific set of abducibles A depends on the application domain.

The adjective *abductive* is used to distinguish explanations in the abductive sense from the notion of explanation in the context of DeLP (or dialectical explanations), where dialectical trees constructed to answer a query can be seen as explanations for the answer obtained. From now on, when we use the term explanation we mean abductive explanation.

After introducing the elements for performing abduction in DeLP, we have to deal with the problem of obtaining explanations. In what follows, we will present a formal characterization of the notion of abductive explanation, in such a way that we could then easily calculate the explanations from this formal characterization.

4 CHARACTERIZING THE NOTION OF (POTENTIAL) ARGUMENT ACTIVATION

As we have seen in section 2, the (state of) warrant of a given literal w.r.t. a DeLP program depends on the arguments that can be computed from that program. Then in order to consider a set Φ of abducibles as an explanation of the warrant of a given literal, Φ must cause the *activation* of certain arguments and, as we will see later, the *non activation* of others. In this section, we formally state the conditions that a given set Φ of abducibles must satisfy in order to cause the activation of a certain argument. More precisely, those conditions specify the presence of certain literals in Φ and the absence of others.

As we have seen in section 2, the notion of argument is defined with respect to a *concrete* DeLP program. On the other hand, when performing abduction we work with an *incomplete* DeLP program, trying to find a set of facts (more precisely abducibles) to *complete* the DeLP program in order to warrant a given literal. Then the notion of argument we introduced in section 2 is not suitable in the context of abduction, and consequently we need to introduce a notion of *potential* argument. For the next definition, we will use $\mathcal{L}(AF)$ to denote the language associated with AF.

Definition 4.1 (Potential Argument). *Let $AF = \langle \Pi_f \cup \Delta, A \rangle$ be an A-DeLP framework. Let $Q \in \mathcal{L}(AF)$. Let $\mathcal{B} \subseteq \Delta$. Then $\langle\langle \mathcal{B}, Q \rangle\rangle$ is a potential argument w.r.t. AF iff there exists a non contradictory set of literals $\Omega \subseteq \mathcal{L}(AF)$ such that $\langle \mathcal{B}, Q \rangle$ is an argument w.r.t. (Ω, Δ).*

Example 4.1. *Consider an A-DeLP framework $AF_1 = \langle \Pi^1_f \cup \Delta_1, A_1 \rangle$, where:*
$\Pi^1_f = \{b, \sim g\}$,
$\Delta_1 = \{(a \prec b, c), (c \prec d, e), (f \prec g), (g \prec h)\}$
$A_1 = \{c, \sim c, d, e, h\}$

Then the following are potential arguments w.r.t. AF_1:

$\langle\langle \{(a \prec b, c), (c \prec d, e)\}, a \rangle\rangle$ *(assuming for example $\Omega = \{b, d, e\}$)*

$\langle\langle \{(a \prec b, c)\}, a \rangle\rangle$ *(assuming for example $\Omega = \{b, c\}$)*

$\langle\langle \emptyset, g \rangle\rangle$ *(assuming for example $\Omega = \{g\}$)*

On the other hand, $\langle\langle \{(a \prec b, c), (f \prec g)\}, a \rangle\rangle$ is not a potential argument w.r.t. AF, since $\langle \{(a \prec b, c), (f \prec g)\}, a \rangle$ will not be an argument w.r.t. (Ω, Δ_1), for any Ω considered. Note that the defeasible rule $f \prec g$ will never be necessary to defeasibly derive a, violating the minimality condition in the definition of argument.

The following definition formally introduces the notion of (potential) argument activation, that was informally used at the beginning of this section.

Definition 4.2 (Potential Argument Activation). *Let $AF = \langle \Pi_f \cup \Delta, A\rangle$ be an A-DeLP framework. Let $\langle\langle \mathcal{B}, Q\rangle\rangle$ be a potential argument w.r.t. AF and let $\Phi \subseteq A$. We say that $\langle\langle \mathcal{B}, Q\rangle\rangle$ is activated by Φ in AF (or alternatively that Φ activates $\langle\langle \mathcal{B}, Q\rangle\rangle$) iff $\Phi \not\vdash \bot$ and $\langle \mathcal{B}, Q\rangle$ is an argument w.r.t. $(\Pi_f \cup \Phi, \Delta)$.*

Example 4.2. *Consider the A-DeLP framework AF_1 of example 4.1. Let $\langle\langle \mathcal{B}, a\rangle\rangle$ be a potential argument w.r.t. AF_1, where $\mathcal{B}$=\{($a \prec b, c$), ($c \prec d, e$)\}. The set of abducibles $\Phi_1 = \{d, e, h\}$ activates $\langle\langle \mathcal{B}, a\rangle\rangle$, because $\langle \mathcal{B}, a\rangle$ is an argument w.r.t. $(\Pi^1_f \cup \Phi_1, \Delta_1)$. As can be seen in figure 2a, Φ_1 completes Π^1_f with those additional facts needed to defeasibly derive a (using rules in $\mathcal{B}$). Furthermore, it can be noted that the abducible $h \in \Phi_1$ does not contribute to the activation of $\langle\langle \mathcal{B}, a\rangle\rangle$. Note that $\Phi_2 = \Phi_1 \setminus \{h\} = \{d, e\}$ also activates $\langle\langle \mathcal{B}, a\rangle\rangle$.*

On the other hand, the set $\Phi_3 = \{d\}$ does not activate $\langle\langle \mathcal{B}, a\rangle\rangle$ (see figure 2b). Note that e does not belong to $\Pi^1_f \cup \Phi_3$, and it is necessary to defeasibly derive a.

The set $\Phi_4 = \{d, e, \sim c\}$ does also not activate $\langle\langle \mathcal{B}, a\rangle\rangle$ (see figure 2c), this time because Φ_4 incorporates a fact ($\sim$c) that contradicts an intermediate conclusion *of $\langle\langle \mathcal{B}, a\rangle\rangle$, violating the consistency condition in the definition of argument.*

Another interesting case is $\Phi_5 = \{d, e, c\}$, which does not activate $\langle\langle \mathcal{B}, a\rangle\rangle$ (see figure 2d). Note that Φ_5 incorporates c, which coincides with the head of $c \prec d, e$, one of the rules in $\mathcal{B}$. In such case, $c \prec d, e$ is unnecessary for deriving a, because as c is a fact, we can directly introduce it in the derivation. So $\mathcal{B}$ is not minimal for deriving a, and then violates the minimality condition of the definition of argument. Now, consider the potential argument $\langle\langle \mathcal{C}, a\rangle\rangle$, where $\mathcal{C} = \{a \prec b, c\}$ (i.e., $\mathcal{C}$ is $\mathcal{B}$ without $c \prec d, e$). Then it holds that Φ_5 effectively activates $\langle\langle \mathcal{C}, a\rangle\rangle$ (see figure 2e).

The set $\Phi_6 = \{e\}$ activates the empty potential argument $\langle\langle \emptyset, e\rangle\rangle$. In general, any set of abducibles Φ containing a literal Q activates the empty potential argument $\langle\langle \emptyset, Q\rangle\rangle$.

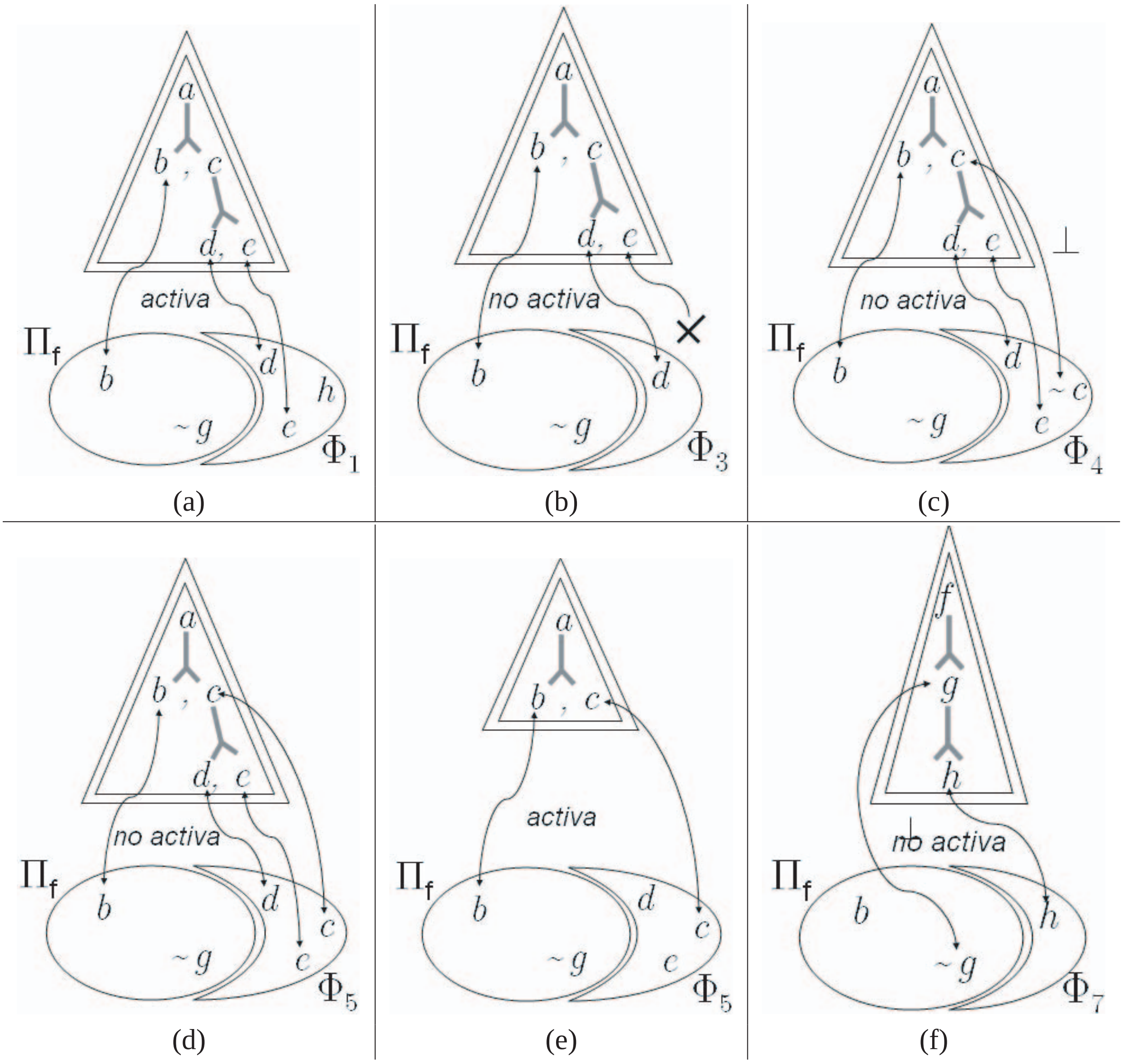
Finally, consider the potential argument $\langle\langle \mathcal{D}, f\rangle\rangle$, where $\mathcal{D} = \{(f \prec g), (g \prec h)\}$, and let $\Phi_7 = \{h\}$. Then Φ_7 does not activate $\langle\langle \mathcal{D}, f\rangle\rangle$ because of an inconsistency involving an intermediate conclusion and a fact (as previously showed with Φ_4 and $\langle\langle \mathcal{B}, a\rangle\rangle$), but this time, the fact causing the inconsistency, $\sim g$, belongs to Π^1_f (see figure 2f). So, there exists no set of abducibles activating $\langle\langle \mathcal{D}, f\rangle\rangle$.

Next, we will introduce a property which states certain necessary and sufficient conditions for a potential argument $\langle\langle \mathcal{B}, Q\rangle\rangle$ to be activated by a set of abducibles Φ. More precisely, the property states which literals must Φ contain (and which must not) in order to activate $\langle\langle \mathcal{B}, Q\rangle\rangle$. Before introducing the property, we need to define two distinguished sets of literals associated with any potential argument.

Definition 4.3. *Let $\langle\langle \mathcal{B}, Q\rangle\rangle$ be a potential argument. We define two sets of literals associated with $\langle\langle \mathcal{B}, Q\rangle\rangle$:*

- $heads(\langle\langle \mathcal{B}, Q\rangle\rangle) =_{def} head\_lits(\mathcal{B})$
- $base(\langle\langle \mathcal{B}, Q\rangle\rangle) =_{def} \begin{cases} \{Q\}, & \textit{if } \mathcal{B} = \emptyset; \\ all\_lits(\mathcal{B}) \setminus head\_lits(\mathcal{B}), & \textit{otherwise.} \end{cases}$

The reason for using of the term *base* becomes evident when considering the graphical representation we adopted for potential arguments in figure 4.2. That is, as a triangle with the conclusion on the top, and expliciting inside the defeasible rules conforming it chained in a top-down derivation. The following example, and the figures associated with it, illustrates what we are saying.



(a) (b) (c)

(d) (e) (f)

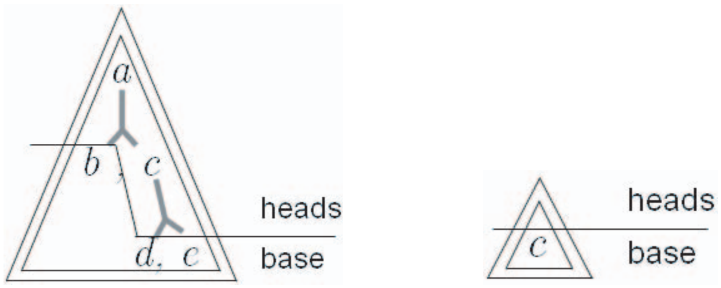
Figure 2: Potential Argument Activation

Example 4.3. *Consider the A-DeLP framework AF_1 of example 4.1. Let $\langle\langle \mathcal{B}, a \rangle\rangle$ be the potential argument of example 4.2 ($\mathcal{B}$=\{($a \prec b, c$), ($c \prec d, e$)\}). Then $heads(\langle\langle \mathcal{B}, a \rangle\rangle) = \{a, c\}$ and $base(\langle\langle \mathcal{B}, a \rangle\rangle) = \{b, d, e\}$ (see figure 3a). Consider the empty potential argument $\langle\langle \emptyset, c \rangle\rangle$. Then $heads(\langle\langle \emptyset, c \rangle\rangle) = \emptyset$ and $base(\langle\langle \emptyset, c \rangle\rangle) = \{c\}$ (see figure 3b).*

Property 4.1. *Let AF =$\langle \Pi_f \cup \Delta, A \rangle$ be an A-DeLP framework. Let $\langle\langle \mathcal{B}, Q \rangle\rangle$ be a potential argument w.r.t. AF and let $\Phi \subseteq A$. Then Φ activates $\langle\langle \mathcal{B}, Q \rangle\rangle$ iff $\Phi \nvdash \perp$ and the following conditions hold:*

- $base(\langle\langle \mathcal{B}, Q \rangle\rangle) \subseteq \Pi_f \cup \Phi$
- *heads(*$\langle\langle \mathcal{B}, Q \rangle\rangle$*)* $\cap (\Pi_f \cup \Phi) = \emptyset$
- $\overline{heads(\langle\langle \mathcal{B}, Q \rangle\rangle)} \cap (\Pi_f \cup \Phi) = \emptyset$

It can easily be verified that the property holds for each case considered in example 4.2. In particular, consider the potential argument $\langle\langle \mathcal{D}, f \rangle\rangle$ defined in example 4.2 ($\mathcal{D}$= \{($f \prec e$), ($e \prec g$)\}).



$\langle\!\langle \mathcal{B}, a \rangle\!\rangle$
(a)

$\langle\!\langle \emptyset, c \rangle\!\rangle$
(b)

Figure 3: base and heads sets of a potential argument.

As we concluded before, there exists no set of abducibles activating $\langle\!\langle \mathcal{D}, f \rangle\!\rangle$. In fact, note that $g \in heads\langle\!\langle \mathcal{D}, f \rangle\!\rangle$ and $\sim\!g \in \Pi_f^1$, and then the second condition in the previous property is violated independently of the set Φ of abducibles considered.

As we have already shown in the previous examples, given an A-DeLP framework, there may exist potential arguments that could not be activated in such framework. That is the same to say, as we have just seen, that they violate at least one of the three conditions of property 4.1, for any set Φ of abducibles considered. So, given an A-DeLP framework, we could ignore such *non activable* potential argument, concentrating only on the others, that we will call *activable* potential arguments. Next, we will formally define the notion of activable potential argument.

Definition 4.4 (Activable Potential Argument). *Let AF =$\langle \Pi_f \cup \Delta, A \rangle$ be an A-DeLP framework. Let $\langle\!\langle \mathcal{B}, Q \rangle\!\rangle$ be a potential argument w.r.t. AF. We say that $\langle\!\langle \mathcal{B}, Q \rangle\!\rangle$ is activable in AF iff there exists a subset Φ of A such that Φ activates $\langle\!\langle \mathcal{B}, Q \rangle\!\rangle$.*

Now that we have formally defined the notion of activable potential argument, we will present, through the next property, a more practical (directly implementable) characterization of that notion. More precisely, it states the restrictions that must satisfy a given potential argument, in terms of it´s base and heads sets, in order to be activable.

Property 4.2. *Let AF =$\langle \Pi_f \cup \Delta, A \rangle$ be an A-DeLP framework. Let $\langle\!\langle \mathcal{B}, Q \rangle\!\rangle$ be a potential argument w.r.t. AF. Then $\langle\!\langle \mathcal{B}, Q \rangle\!\rangle$ is activable in AF (noted as $\langle\!\langle \mathcal{B}, Q \rangle\!\rangle^+$) iff it satisfies the following conditions:*

- $base(\langle\!\langle \mathcal{B}, Q \rangle\!\rangle) \subseteq \Pi_f \cup A$
- $heads(\langle\!\langle \mathcal{B}, Q \rangle\!\rangle) \cap \Pi_f = \emptyset$
- $\overline{heads(\langle\!\langle \mathcal{B}, Q \rangle\!\rangle)} \cap \Pi_f = \emptyset$

Consider the three arguments that appear in example 4.2. Note that the potential argument $\langle\!\langle \mathcal{D}, f \rangle\!\rangle$ ($\mathcal{D} = \{(f \prec e), (e \prec g)\}$), which was identified as non-activable, violates indeed the third condition of this property. The other two, $\langle\!\langle \mathcal{B}, a \rangle\!\rangle$ ($\mathcal{B}$={($a \prec b, c$), ($c \prec d, e$)}) and $\langle\!\langle \mathcal{C}, a \rangle\!\rangle$, ($\mathcal{C} = \{a \prec b, c\}$), are activable (see figures 2a and 2e respectively) and then as it can be easily verified, they both satisfy the three conditions of the property.

Finally, we will present a new version of property 4.1, specific for activable potential arguments.

Property 4.3. *Let AF =$\langle \Pi_f \cup \Delta, A \rangle$ be an A-DeLP framework. Let $\langle\!\langle \mathcal{B}, Q \rangle\!\rangle^+$ be an activable potential argument w.r.t. AF and let $\Phi \subseteq A$. Then Φ activates $\langle\!\langle \mathcal{B}, Q \rangle\!\rangle^+$ iff $\Phi \nvdash \bot$ and the following conditions are satisfied:*

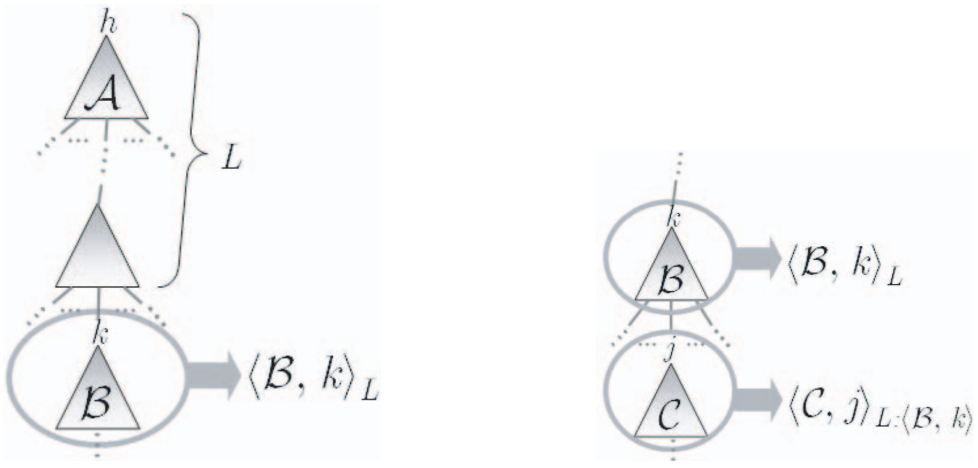
- $base(\langle\!\langle \mathcal{B}, Q \rangle\!\rangle^{+}) \setminus \Pi_f \subseteq \Phi$
- $heads(\langle\!\langle \mathcal{B}, Q \rangle\!\rangle^{+}) \cap \Phi = \emptyset$
- $\overline{heads(\langle\!\langle \mathcal{B}, Q \rangle\!\rangle^{+})} \cap \Phi = \emptyset$

5 CHARACTERIZING WARRANT EXPLANATIONS

As we have said in section 3, our purpose is to provide a formal characterization of the notion of abductive explanation, in such a way that we could then easily calculate the explanations from this formal characterization. In the previous section, we presented partially this characterization, focusing only on the conditions under which a given set Φ of abducibles activates a given (activable) potential argument. In this section we will complete the characterization. More specifically, we will provide a logical formalization in First Order Logic that captures the notion of which (activable) potential arguments must a given set Φ of abducibles activate, and which must not, in order to explain the warrant of a given literal h.

As a first step, we will define a specific notation for denoting nodes in a dialectical tree. At first sight, it seems that to univocally identify a certain node it is sufficient to name the argument labelling it. But that is wrong because two nodes in a dialectical tree could be labelled with the same argument. An effective way of identifying a node is making reference not only to the argument labelling it, but also to the argumentation line conformed by the (arguments labelling the) ancestors of the node in the tree.

Consider a dialectical tree $\mathcal{T}(\langle \mathcal{A}, h \rangle)$, let N be a node of $\mathcal{T}(\langle \mathcal{A}, h \rangle)$ labeled with $\langle \mathcal{B}, k \rangle$ and let L be the argumentation line corresponding to the path from the root of $\mathcal{T}(\langle \mathcal{A}, h \rangle)$ to the father of N inclusively. Then we will use $\langle \mathcal{B}, k \rangle_L$, read as $\langle \mathcal{B}, k \rangle$ with ancestors line L, to refer to the node N (see figure 4a). Additionally, let $L = [\langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle, ..., \langle \mathcal{A}_n, h_n \rangle]$ be an arbitrary argumentation line, and let $\langle \mathcal{B}, k \rangle$ be an argument. Then will use $L : \langle \mathcal{B}, k \rangle$ to denote the argumentation line $[\langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle, ..., \langle \mathcal{A}_n, h_n \rangle, \langle \mathcal{B}, k \rangle]$, that results from adding $\langle \mathcal{B}, k \rangle$ at the end of L. Finally, if $\langle \mathcal{B}, k \rangle_L$ denotes a node and it has a child labeled with $\langle \mathcal{C}, j \rangle$, then combining both notations introduced, we can use $\langle \mathcal{C}, j \rangle_{L:\langle \mathcal{B}, k \rangle}$ to denote the child node (see figure 4b).



(a) (b)

Figure 4: Notation for nodes in a dialectical tree.

Next we will present the analysis that lead us to the first logic formula. Let $AF = \langle \mathcal{P}, A \rangle$ be an A-DeLP framework, let $\Phi \subseteq A$ be a consistent set of abducibles and let h be a literal. Then from the definition of warrant presented in Section 2 and from our previous discussion we can state the following:

h will be warranted w.r.t. the DeLP program $\mathcal{P} \cup \Phi$ (1)

iff

there exist an argument $\langle \mathcal{A}, h \rangle$ in $\mathcal{P} \cup \Phi$ (2)

and

$\mathcal{T}_{(\mathcal{P} \,\cup\, \Phi)}(\langle \mathcal{A}, h \rangle)$, the dialectical tree for $\langle \mathcal{A}, h \rangle$, has its root undefeated.

But the statement (1) is the same to say that Φ *will be an explanation for the warrant of h.* Also, the statement (2) is in turn the same to say that *there exist an activable potential argument $\langle\langle \mathcal{A}, h \rangle\rangle^+$ in AF such that Φ activates $\langle\langle \mathcal{A}, h \rangle\rangle^+$.* By rephrasing statements (1) and (2) into the equivalent formulation given before, we can define the notion of explanation as follows:

*Φ will be an explanation for the warrant of h **iff** there exists an activable potential argument $\langle\langle \mathcal{A}, h \rangle\rangle^+$ in AF **such that** Φ activates $\langle\langle \mathcal{A}, h \rangle\rangle^+$ **and** $\mathcal{T}_{(\mathcal{P} \,\cup\, \Phi)}(\langle \mathcal{A}, h \rangle)$ has its root undefeated.*

Let $PotArgs^+_{AF}$ be the set of all activable potential arguments w.r.t. AF. Then the previous sentence is captured by the following formula:

$$exp\_warrant(\Phi, h) =_{def} \bigvee_{\langle\langle \mathcal{A}, h \rangle\rangle^+ \in \; PotArgs^+_{AF}} act(\Phi, \langle\langle \mathcal{A}, h \rangle\rangle^+) \;\wedge\; U(\Phi, \langle \mathcal{A}, h \rangle_{[\,]}) \qquad (\alpha)$$

Now, consider an argument $\langle \mathcal{A}, h \rangle$ w.r.t. $\mathcal{P} \cup \Phi$, let $\mathcal{T}_{(\mathcal{P} \,\cup\, \Phi)}(\langle \mathcal{A}, h \rangle)$ be the dialectical tree for $\langle \mathcal{A}, h \rangle$ and let $\langle \mathcal{N}, k \rangle_L$ be a node of $\mathcal{T}_{(\mathcal{P} \,\cup\, \Phi)}(\langle \mathcal{A}, h \rangle)$. Then according to the criterion used for marking a dialectical tree (section 2), we can say that:

*$\langle \mathcal{N}, k \rangle_L$ will be defeated **iff** there exists a child $\langle \mathcal{M}, q \rangle_{L:\langle \mathcal{N}, k \rangle}$ of $\langle \mathcal{N}, k \rangle_L$* (3)
***such that** $\langle \mathcal{M}, q \rangle_{L:\langle \mathcal{N}, k \rangle}$ is undefeated.*

But, by definition of dialectical tree, the node $\langle \mathcal{N}, k \rangle_L$ has a child $\langle \mathcal{M}, q \rangle_{L:\langle \mathcal{N}, k \rangle}$ iff there exists an argument $\langle \mathcal{M}, q \rangle$ w.r.t. $\mathcal{P} \cup \Phi$ which extends the argumentation line $L : \langle \mathcal{N}, k \rangle$. Moreover, that is equivalent to say that Φ activates some potential argument $\langle\langle \mathcal{M}, q \rangle\rangle^+$ w.r.t. AF which extends $L : \langle \mathcal{N}, k \rangle$. Below is the statement that results from transcribing (3) according to de last analysis:

*$\langle \mathcal{N}, k \rangle_L$ will be defeated **iff** there exists an activable potential argument $\langle\langle \mathcal{M}, q \rangle\rangle^+$ w.r.t. AF which extends $L : \langle \mathcal{N}, k \rangle$ **such that** Φ activates $\langle\langle \mathcal{M}, q \rangle\rangle^+$ **and** $\langle \mathcal{M}, q \rangle_{L:\langle \mathcal{N}, k \rangle}$ is undefeated.*

OBSERVATION: We could restrict $\langle\langle \mathcal{M}, q \rangle\rangle^+$ to be a non-empty argument without changing the meaning of the sentence. The reason is that if $\mathcal{M} = \emptyset$, we can ensure that $\langle\langle \mathcal{M}, q \rangle\rangle^+$ will not be activated by Φ [1].

Let L be an argumentation line. We will use $PotArgsExt^+_{AF}(L)$ to denote the set of all non-empty activable potential arguments w.r.t. AF which extend L. Then the previous sentence is captured by the following formula:

$$D(\Phi, \langle \mathcal{N}, k \rangle_L) =_{def} \bigvee_{\langle\langle \mathcal{M}, q \rangle\rangle^+ \in \; PotArgsExt^+_{AF}(L:\langle \mathcal{N}, k \rangle)} act(\Phi, \langle\langle \mathcal{M}, q \rangle\rangle^+) \;\wedge\; U(\Phi, \langle \mathcal{M}, q \rangle_{L:\langle \mathcal{N}, k \rangle}) \qquad (\beta)$$

Making a similar analysis, but this time for the *undefeated* status of a node in a dialectical tree, we reach to the following sentence:

[1] Note that since $\langle \mathcal{N}, k \rangle$ is an argument w.r.t. $\mathcal{P} \cup \Phi$, by consistency condition of the definition of argument, we know that there is no strong information (in $\mathcal{P} \cup \Phi$) contradicting $\langle \mathcal{N}, k \rangle$, which in turn implies that there exist no empty argument $\langle \emptyset, q \rangle$ in $\mathcal{P} \cup \Phi$ contradicting $\langle \mathcal{N}, k \rangle$.

$\langle \mathcal{N}, k\rangle_L$ will be defeated ***iff*** *for all (non empty) activable potential argument $\langle\langle \mathcal{M}, q\rangle\rangle^+$ w.r.t. AF which extends $L : \langle \mathcal{N}, k\rangle$, it holds that* ***either*** *Φ does not activate $\langle\langle \mathcal{M}, q\rangle\rangle^+$* ***or*** *$\Phi$ does activate $\langle\langle \mathcal{M}, q\rangle\rangle^+$* ***and also*** *$\langle \mathcal{M}, q\rangle_{L:\langle \mathcal{N},k\rangle}$ is defeated.*

The next formula captures the meaning of the previous sentence:

$$U(\Phi, \langle \mathcal{N}, k\rangle_L) =_{def} \qquad (\gamma)$$

$$\bigwedge_{\langle\langle \mathcal{M},q\rangle\rangle^+ \in PotArgsExt^+_{AF}(L:\langle \mathcal{N},k\rangle)} \neg act(\Phi, \langle\langle \mathcal{M}, q\rangle\rangle^+) \vee \big(act(\Phi, \langle\langle \mathcal{M}, q\rangle\rangle^+) \wedge D(\Phi, \langle \mathcal{M}, q\rangle_{L:\langle \mathcal{N},k\rangle})\big)$$

Finally, we will present an example illustrating how the previous formulas can be used.

Example 5.1. *Consider an A-DeLP framework $AF_2 = \langle \Pi^2_f \cup \Delta_2, A_2\rangle$, where*
$\Pi^2_f = \{e, b\}$
$\Delta_2 = \{(a \prec b, c), (\sim a \prec d), (\sim a \prec g), (d \prec e), (g \prec c), (\sim d \prec e, f)\}$
$A_2 = \{c, \sim d, f, \sim g\}$

Next, we list all the (non empty) activable potential arguments w.r.t. AF_2:

$\langle\langle A_1, a\rangle\rangle^+$, *where* $A_1 = \{a \prec b, c\}$
$\langle\langle A_2, \sim a\rangle\rangle^+$, *where* $A_2 = \{(\sim a \prec d), (d \prec e)\}$
$\langle\langle A_3, \sim d\rangle\rangle^+$, *where* $A_3 = \{\sim d \prec e, f\}$
$\langle\langle A_4, \sim a\rangle\rangle^+$, *where* $A_4 = \{(\sim a \prec g), (g \prec c)\}$

Suppose we want to explain the warrant of a in AF_2. Then we need first to instantiate (and generate) the parametric formula (α) for AF_2 and taking h as a. The instantiated formula follows:

$$exp\_warrant(\Phi, a) = act(\Phi, \langle\langle A_1, a\rangle\rangle^+) \;\wedge\; U(\Phi, \langle A_1, a\rangle_{[\,]})$$

But the atom $U(\Phi, \langle A_1, a\rangle_{[\,]})$ in the previous instantiated formula denotes also an entire formula, that could be obtained instantiating (γ). The resulting instantiated formula follows next:

$$\begin{aligned} U(\Phi, \langle A_1, a\rangle_{[\,]}) = \quad & \neg act(\Phi, \langle\langle A_2, \sim a\rangle\rangle^+) \;\vee\; \big(act(\Phi, \langle\langle A_2, \sim a\rangle\rangle^+) \;\wedge\; D(\Phi, \langle A_2, \sim a\rangle_{[\langle A_1,a\rangle]})\big) \\ & \wedge \\ & \neg act(\Phi, \langle\langle A_4, \sim a\rangle\rangle^+) \;\vee\; \big(act(\Phi, \langle\langle A_4, \sim a\rangle\rangle^+) \;\wedge\; D(\Phi, \langle A_4, \sim a\rangle_{[\langle A_1,a\rangle]})\big) \end{aligned}$$

Now two new atoms have been introduced, this time denoting formulas that can be obtained instantiating formula (β). We keep on doing that until no more atoms are introduced. The remaining instantiated formulae follows next:

$$D(\Phi, \langle A_2, \sim a\rangle_{[\langle A_1,a\rangle]}) = act(\Phi, \langle\langle A_3, \sim d\rangle\rangle^+) \;\wedge\; U(\Phi, \langle A_3, \sim d\rangle_{[\langle A_1,a\rangle, \langle A_2, \sim a\rangle]})$$

$$D(\Phi, \langle A_4, \sim a\rangle_{[\langle A_1,a\rangle]}) = false$$

$$U(\Phi, \langle A_3, \sim d\rangle_{[\langle A_1,a\rangle, \langle A_2, \sim a\rangle]}) = true$$

Finally, starting form the instantiated formula for $exp\_warrant(\Phi, a)$ and substituting each D and U atom by its definition, we obtain the following formula, in which only act/1 atoms occur:

$$\begin{aligned} exp\_warrant(\Phi, a) = \quad & act(\Phi, \langle\langle A_1, a\rangle\rangle^+) \\ & \wedge \\ & \Big(\neg act(\Phi, \langle\langle A_2, \sim a\rangle\rangle^+) \;\vee\; \big(act(\Phi, \langle\langle A_2, \sim a\rangle\rangle^+) \wedge act(\Phi, \langle\langle A_3, \sim d\rangle\rangle^+)\big)\Big) \;\wedge \neg act(\Phi, \langle\langle A_4, \sim a\rangle\rangle^+) \end{aligned}$$

That formula clearly specifies which activable potential arguments must Φ activate and which must not in order to be an explanation for the warrant of a. Let´s analyze what is concretely saying.

Note that the $\vee$ in the formula reflect alternative restrictions. In fact, the formula describes two alternative situations on which Φ constitutes an explanation for a. One is that Φ activates $\langle\langle A_1, a\rangle\rangle^+$ and does not activate $\langle\langle A_2, {\sim}a\rangle\rangle^+$ nor $\langle\langle A_4, {\sim}a\rangle\rangle^+$. The other is that Φ activates $\langle\langle A_1, a\rangle\rangle^+$, $\langle\langle A_2, {\sim}a\rangle\rangle^+$ and $\langle\langle A_3, {\sim}d\rangle\rangle^+$ and does not activate $\langle\langle A_4, {\sim}a\rangle\rangle^+$. The dialectical trees warranting a associated to each alternative are shown in figures 5a and 5b.

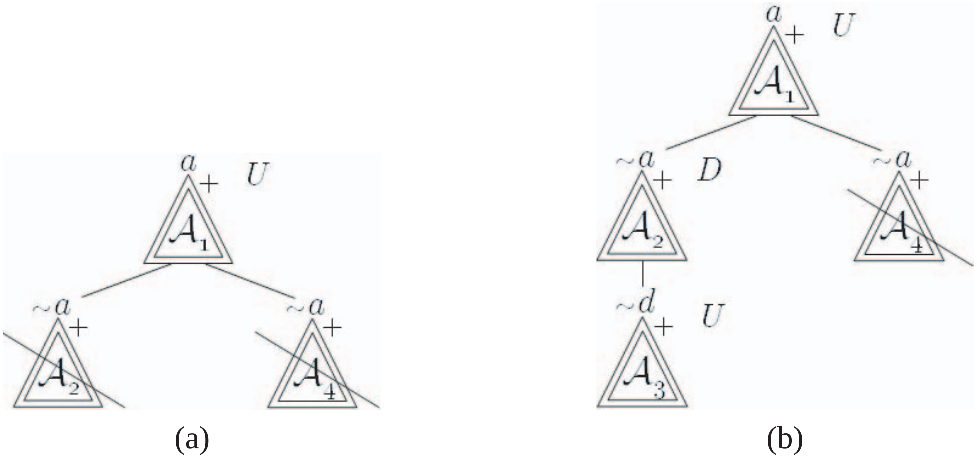


Figure 5: Alternative dialectical trees warranting a.

6 CONCLUSIONS AND FUTURE WORK

In this article, we presented the confluence of two general ideas: Defeasible Logic Programming and Abduction in logic. Concretely, we introduced a framework that formally states the problem of doing abduction (obtaining abductive explanations) in the particular case that the abductive theory is a DeLP Program. In the reminder of this work, we faced that problem, providing a formal characterization of the notion of abductive explanation, from which we could then easily calculate the explanations. That characterization was presented in two parts. On the one hand, we formally stated the conditions that must satisfy an explanation in order to activate a given argument. On the other hand, we formally stated which arguments must an explanation activate and which must not in order to effectively explain the warrant of a literal.

As future work, we will develop a procedure for efficiently calculating the explanations. That procedure will be tightly based on the formal characterization presented here. Then as we have outlined in this article, by proving the correctness of the characterization, we cold ensure the correctness of the procedure.

REFERENCES

[1] Delp web page: http://lidia.cs.uns.edu.ar/delp.

[2] A. García and G. Simari. Defeasible logic programming: An argumentative approach. *Theory Practice of Logic Programming*, 4(1):95–138, 2004.

[3] Vladimir Lifschitz. Foundations of logic programs. In G. Brewka, editor, *Principles of Knowledge Representation*, pages 69–128. CSLI Pub., 1996.