A Preliminary Investigation on a Revision-Based Approach to the Status of Warrant*

Martín O. Moguillansky
Nicolás D. Rotstein
Marcelo A. Falappa
Guillermo R. Simari

Consejo de Investigaciones Científicas y Técnicas (CONICET)
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)
Departamento de Ciencias e Ingeniería de la Computación (DCIC)
Universidad Nacional del Sur (UNS)
Av. Alem 1253 - (B8000CPB) Bahía Blanca - Argentina
PHONE/FAX: (+54)(291)459-5136
E-MAIL: {mom, ndr, mfalappa, grs}@cs.uns.edu.ar

Abstract

In this article we are presenting a new perspective on the matter of belief revision by its relation to argumentation systems. Our approach is based on the argumentative formalism Defeasible Logic Programming, and therefore we propose a revision of a defeasible logic program by an argument. The revision operators here introduced are defined as prioritized, since they ensure warrant of the conclusion of the argument being added to the program following a particular minimal change principle. To achieve this, we give two different approaches: one regarding arguments in the classical sense, whereas the other considers the revision by arguments that also include strict rules and facts. Finally, a brief discussion about the relation between our approach and the basic theory of belief revision is exposed, along with a description of other possible minimal change principles.

Keywords: belief revision, argumentation, defeasible logic programming, non-monotonic reasoning.

1 Introduction & Motivation

This work offers a first approach to revision in argumentation systems, using Defeasible Logic Programming (DELP) as base formalism. The objective is to define an **argument revision operator** that ensures warrant of the conclusion of the (external) argument being added to a program. In that sense, this operator will be **prioritized**. When we revise a program $\mathcal{P}$ by an argument $\langle \mathcal{A}, \alpha \rangle$, the program resulting from the revision will be such that $\mathcal{A}$ is an undefeated argument and α is then warranted. Because of this, we named the operator **warrant-prioritized revision operator**.

*This article assumes background knowledge on argumentation and belief revision from the reader.

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 main issue underlying warrant-prioritized argument revision lies in the selection of arguments and the incisions that have to be made over them. Incisions will make these arguments "disappear", but selections have to be done carefully, following some minimal change principle. In this work, we present one minimal change principle, thus defining a warrant-prioritized revision operator. The corresponding selection and incision functions are defined, along with some properties it should verify. This revision operator is therefore reconsidered, by applying it to extended arguments.

The article is structured as follows: Section 2 gives an overview of the main concepts involved in the DELP formalism, Section 3 describes the notions of the belief revision theory we inspired from to define this approach, Section 4 explains in detail the two versions of the revision operator, and Section 5 gives a final discussion on revision in argument systems and poses future lines of work.

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 [6] 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 the 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. 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 is used to represent 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 [12], some examples use "schematic rules" with variables. To distinguish variables, as usual, they start with an uppercase letter.

Example 1 *Consider the* DELP*-program* (Π_1, Δ_1) *where:*

$$\Pi_1 = \left\{ \begin{array}{ll} (bird(X) \leftarrow chicken(X)) & chicken(little) \\ chicken(tina) & scared(tina) \end{array} \right\}$$

$$\Delta_1 = \left\{ \begin{array}{l} flies(X) -\!\prec bird(X) \\ flies(X) -\!\prec chicken(X), scared(X) \\ \sim flies(X) -\!\prec chicken(X) \end{array} \right\}$$

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.

From a program is possible to derive contradictory literals, *e.g.*, from (Π_1, Δ_1) of Ex. 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 the 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 [6]).

Definition 1 (Argument Structure) *Let* (Π, Δ) *be a* DELP*-program,* $\langle \mathcal{A}, \alpha \rangle$ *is an argument structure 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 *From the* DELP*-program* (Π_1, Δ_1) *the following arguments can be obtained:*
$\langle \mathcal{A}_1, flies(tina) \rangle = \langle \{ flies(tina) \text{–}\prec bird(tina) \}, flies(tina) \rangle$
$\langle \mathcal{A}_2, \sim flies(tina) \rangle = \langle \{ \sim flies(tina) \text{–}\prec chicken(tina) \}, \sim flies(tina) \rangle$
$\langle \mathcal{A}_3, flies(tina) \rangle = \langle \{ flies(tina) \text{–}\prec chicken(tina), scared(tina) \}, flies(tina) \rangle$

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 this 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. 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 [6]).

Example 3 *From Ex. 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 arguments sequence* $[\langle \mathcal{A}_1, flies(tina) \rangle, \langle \mathcal{A}_2, \sim flies(tina) \rangle, \langle \mathcal{A}_3, flies(tina) \rangle]$ *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 one argument, leading to a tree structure.

Definition 2 (Dialectical tree [6]) *A dialectical tree for* $\langle \mathcal{A}_0, h_0 \rangle$*, denoted* $\mathcal{T}(\langle \mathcal{A}_0, h_0 \rangle)$*, 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 sequence of labels of the path from the root to* N*.*
Let $\{ \langle \mathcal{B}_1, q_1 \rangle, \langle \mathcal{B}_2, q_2 \rangle, \ldots, \langle \mathcal{B}_k, q_k \rangle \}$ *be the set of all the defeaters for* $\langle \mathcal{A}_n, h_n \rangle$*. For each defeater* $\langle \mathcal{B}_i, q_i \rangle$ $(1 \leq i \leq k)$*, such that, the argumentation line* $\Lambda' = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle, \langle \mathcal{B}_i, q_i \rangle]$ *is acceptable, then the node* N *has a child* N_i *labelled* $\langle \mathcal{B}_i, q_i \rangle$*.*
If there is no defeater for $\langle \mathcal{A}_n, h_n \rangle$ *or there is no* $\langle \mathcal{B}_i, q_i \rangle$ *such that* Λ' *is acceptable, 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 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 $\mathcal{T}(\langle \mathcal{A}, h \rangle)$ 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 $\langle \mathcal{B}, q\rangle$ be an inner node of $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h\rangle)$. Then $\langle \mathcal{B}, q\rangle$ will be marked as "U" iff every child of $\langle \mathcal{B}, q\rangle$ is marked as "D". The node $\langle \mathcal{B}, q\rangle$ 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 $\mathcal{P}$.

In this paper, marked dialectical trees will be depicted as a tree of triangles where edges denote the defeat relation (in Figure 1, three marked dialectical trees are shown). An argument $\langle \mathcal{A}, h\rangle$ will be depicted as a triangle, where its upper vertex is labelled with the conclusion h, and the set of defeasible rules $\mathcal{A}$ are associated with the triangle itself. Gray triangles will be undefeated arguments, whereas white triangles will depict defeated arguments. In the rest of the article we will refer to marked dialectical trees just as "D-tree".

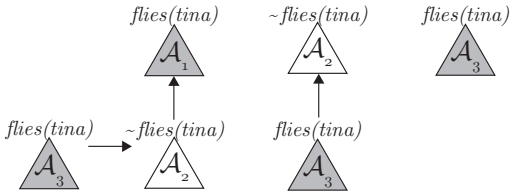


Figure 1: D-trees for $flies(tina)$

Example 4 *(Extends Ex. 3) Figure 1 shows the D-tree for $\mathcal{T}^*_{\mathcal{P}_1}(\langle \mathcal{A}_1, f\rangle)$ (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 $\mathcal{T}^*_{\mathcal{P}_1}(\langle \mathcal{A}_1, f\rangle)$ is marked as "U" and therefore the literal 'flies(tina)' is warranted.*

3 Belief Revision Overview

A **belief base** is a knowledge state represented by a set of sentences not necessarily closed under logical consequence. A **belief set** is a set of sentences in a given language, closed under logical consequence. In general, a belief set is infinite being this the main reason of the impossibility to deal with this kind of sets in a computer. Instead, it is possible to characterize the properties that each of the change operations should satisfy on any finite representation of a knowledge state.

Classic operations in the theory change [2] are known as expansions, contractions, and revisions. An **Expansion** operation noted with "$+$", adds a new belief to the epistemic state, without guaranteeing its consistency after the operation. A **Contraction** operation, noted with "$-$", eliminates a belief α from the epistemic state and those beliefs that make possible its deduction or inference. The sentences to eliminate might represent the *minimal change* on the epistemic state. Finally, a **Revision** operation ("$*$") inserts sentences to the epistemic state, guaranteeing consistency (if it was consistent before the operation). This means that a revision adds a new belief and perhaps it eliminates other ones in order to avoid inconsistencies.

Other non-classical operations exists, like **Merge** [13] noted with " $\circ$ ", which fusions belief bases or sets assuring a consistent resultant epistemic state, and **Consolidations** (" ! ") that restore consistency to a contradictory epistemic state. Usually, slight extensions or modifications of these operations are needed in order to capture different improved features of the environment required to work

in. This is the case of the operator " $\diamond$ " used to represent a **Kernel Revision by a Set of Sentences** [4] operation, which defines a non-prioritized version of a kernel revision operation, providing the knowledge to be revised and its logical proof (or argument) given by a consistent set of sentences.

3.1 Kernel Contractions

The **Kernel Contraction** operator, applicable to belief bases and belief sets, consists of a contraction operator capable of the selection and elimination of those beliefs in K that contribute to infer α.

Definition 3 (Set of Kernels [9]) *Let K be a set of sentences and α a sentence. The set $K^{\perp\!\!\!\perp}\alpha$, called* set of kernels *is the set of sets K' such that* (1) $K' \subseteq K$, (2) $K' \vdash \alpha$, *and* (3) *if $K'' \subset K'$ then $K'' \nvdash \alpha$. The set $K^{\perp\!\!\!\perp}\alpha$ is also called* set of α-kernels *and each one of its elements are called* α-kernel.

For the success of a contraction operation we need to eliminate, at least, one element of each α-kernel. The elements to be eliminated are selected by an **Incision Function**.

Definition 4 (Incision Function [9]) *Let K be a set of sentences and "σ" be an* incision function *for it such that for any sentence α it verifies,* (1) $\sigma(K^{\perp\!\!\!\perp}\alpha) \subseteq \bigcup(K^{\perp\!\!\!\perp}\alpha)$ *and* (2) *If $K' \in K^{\perp\!\!\!\perp}\alpha$ and $K' \neq \emptyset$ then $K' \cap \sigma(K^{\perp\!\!\!\perp}\alpha) \neq \emptyset$.*

Once the incision function was applied we must eliminate from K those sentences that the incision has selected, *i.e.*, the new belief base would consist of all those sentences that kept outside of the scope of σ.

Definition 5 (Kernel Contraction Determined by "σ" [9]) *Let K be a set of sentences, α a sentence, and $K^{\perp\!\!\!\perp}\alpha$ the set of α-kernels of K. Let "σ" be an incision function for K. The operator " $-_\sigma$ ", called* kernel contraction determined by "σ", *is defined as, $K -_\sigma \alpha = K \backslash \sigma(K^{\perp\!\!\!\perp}\alpha)$.*

Finally, an operator " $-$ " is a kernel contraction operator for K if and only if there exists an incision function "σ" such that $K - \alpha = K -_\sigma \alpha$ for all sentence α.

3.2 Consistent Incorporation of Beliefs

A revision operator "$*$" looks for the addition of the new belief α to the belief set K, and therefore the assurance that the resulting belief set $K * \alpha$ is consistent (unless α is inconsistent). The first task can be accomplished by expansion by α. The second can be accomplished by prior contraction by its negation $\sim\alpha$. If a belief set does not imply $\sim\alpha$, then α can be added to it without loss of consistency. This composition of sub-operations gives rise to the following definition of a revision operator [7, 11]:

$$\textbf{(Levi Identity)}\ K * \alpha = (K - \sim\alpha) + \alpha$$

We will define the revision operation in a set K regarding a sentence α, by means of the *Levi Identity*, assuming that " $-$ " is a *kernel contraction* operator determined by an incision function "σ".

Definition 6 (Internal (External) Kernel Revision [10]) *Let " $-$ " be a kernel contraction for a set K. Then the **Internal Kernel Revision** operator for K is defined as $K \mp_\sigma \alpha = (K - \sim\alpha) + \alpha$. Analogously, an **External Kernel Revision** operator is defined as $K \pm_\sigma \alpha = (K + \alpha) - \sim\alpha$.*

Finally, a kernel revision operator "$*$" may be characterized by either an internal "$\mp_\sigma$" or an external "$\pm_\sigma$" kernel revision.

3.3 Non Prioritized Revisions

The classic revision operation is characterized by the postulates of rationality introduced by Gärdenfors [8], some of them have been argued for being considered arbitraries. Particularly, the *success* postulate ($\alpha \in K * \alpha$) establishes that a new information to be revised in an epistemic state must be part of it, despite that other beliefs in the agent's state must be eliminated in order to maintain its consistency. For that purpose is interesting to define new types of revision operations to "catch" the information in a more intuitively way such that a new information has "*no absolute priority*" over those in the epistemic state.

Definition 7 (Explanation Set [5]) *The set A is an explanation for α iff it means a **minimal** proof for α, it is consistent and it is not self-explanatory, i.e., $\alpha \nvdash A$.*

Usually one does not totally accept what others inform but only what one considers to be relevant. This property is known as *partial acceptation*, and its behavior may be modeled by a multiple revision operator as follows:

Definition 8 (Non-Prioritized Multiple Revision [5]) *Let "σ" be an incision function, and let K and A be two sets of sentences, such that K is consistent and A finite. The **Non-Prioritized** Kernel Revision by a Set of Sentences operator "$\diamond_{\mathcal{N}}$" is defined as follows:*

$$K \diamond_{\mathcal{N}} A = (K \cup A) \backslash \sigma((K \cup A)^{\perp\!\!\!\perp} \perp)$$

4 Argument Revision Operators

Intuitively, a "**Warrant-Prioritized Argument Revision Operator**" (for short: WP Argument Revision Operator) revises a given program $\mathcal{P} = (\Pi, \Delta)$ by an external argument $\langle \mathcal{A}, \alpha \rangle$. Moreover, this argument ends up being warranted from the program resulting from the revision, provided that $\mathcal{A} \cup \Pi$ has a defeasible derivation for α. The set Π of strict rules and facts represents (in a way) the current state of the world. The external argument $\langle \mathcal{A}, \alpha \rangle$ provides a set of defeasible rules that jointly with the state of the world decides in favor of the conclusion α, *i.e.*, it poses a reason to believe in it. Hence, this argument does not stand by itself, but in conjunction with the strict part of the program it is being added to, *i.e.*, α is defeasibly derived from $\mathcal{A} \cup \Pi$.

Although it would be interesting to revise a program by $\langle \mathcal{A}, \alpha \rangle$ only when α is not already warranted (by another argument), it might be desirable to have $\mathcal{A}$ as an undefeated argument. In our approach, we take this last posture: the WP Argument Revision Operator will ensure $\mathcal{A}$ to be an undefeated argument. In this way, α would be always warranted.

For this matter, a *hypothetical dialectical tree* rooted on $\langle \mathcal{A}, \alpha \rangle$ is built. The D-tree is deemed as *"hypothetical"* due to $\langle \mathcal{A}, \alpha \rangle$ not belonging to $\mathcal{P}$. Incisions over arguments in this tree are made in order to turn $\mathcal{A}$ into an undefeated argument. Selections (consequently, incisions) must agree with some minimal change principle. In this work, we propose a principle that attempts to ensure minimal deletion of the DELP-program rules.

Finally, in the examples given throughout the article, we abstract away the argument comparison criterion: we will just give DELP-programs, pointing out which is the associated D-tree for the argument being added, and thereafter the analysis begins.

4.1 WP Argument Revision – Minimal Change wrt. the DeLP-program

A WP Argument Revision Operator "$*^{\omega}_{\mathcal{P}}$" attempts to insert an argument $\langle \mathcal{A}, \alpha \rangle$ into a program $\mathcal{P}$, in such a way that α turns out to be warranted from $\mathcal{P} *^{\omega}_{\mathcal{P}} \langle \mathcal{A}, \alpha \rangle$. Revising a program $\mathcal{P} = (\Pi, \Delta)$ by an argument $\langle \mathcal{A}, \alpha \rangle$ involves the generation of a hypothetical D-tree rooted in $\langle \mathcal{A}, \alpha \rangle$, namely $\mathcal{T}^{*}_{\mathcal{P}'}(\langle \mathcal{A}, \alpha \rangle)$, where $\mathcal{P}' = (\Pi, \Delta \cup \mathcal{A})$. Therefore, since we want α to be warranted, those undefeated defeaters for $\langle \mathcal{A}, \alpha \rangle$ will be *cut off* in order to turn $\langle \mathcal{A}, \alpha \rangle$ into an undefeated argument.

Definition 9 (Argument Selection Function "$\gamma^{\omega}_{\mathcal{P}}$") *Let $T = \mathcal{T}^{*}_{\mathcal{P}'}(\langle \mathcal{A}, \alpha \rangle)$ be a* D-tree *and λ_i a dialectical line rooted in $\langle \mathcal{A}, \alpha \rangle$, then $\gamma^{\omega}_{\mathcal{P}}(\lambda_i) = \mathcal{B}$ iff $\mathcal{B}$ is a defeater for $\langle \mathcal{A}, \alpha \rangle$ marked as* **undefeated** *in T. From now on, the argument selected in the i^{th} dialectical line will be called Ψ_i.*

In general, selecting defeaters for the root argument ensures a minimal deletion of defeasible rules from the DELP-program at issue. That is because the deletion of a root's defeater eliminates a whole branch. Trying to achieve the same result by deleting rules from "lower" arguments would affect a greater amount of arguments, due to the possibility of branching.

Definition 10 (Argument Incision Function "$\sigma^{\omega}_{\mathcal{P}}$") *Let Ψ_i be the* ***"less relevant interference argument"*** *determined by an argument selection function $\gamma^{\omega}_{\mathcal{P}}$ in the dialectical line λ_i. Then a function $\sigma^{\omega}_{\mathcal{P}}$ is an* argument incision function *iff it verifies $\emptyset \subset \sigma^{\omega}_{\mathcal{P}}(\Psi_i) \subseteq \Psi_i$.*

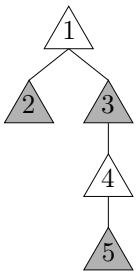
Example 5 *Let consider a program $\mathcal{P}_5 = (\Pi_5, \Delta_5)$, where:*

$$\Pi_5 = \left\{ \begin{array}{l} t, \\ z \end{array} \right\} \quad \Delta_5 = \left\{ \begin{array}{l} \sim a \prec y, y \prec x, \sim a \prec z, \\ a \prec w, w \prec y, \sim a \prec t \end{array} \right\}$$

From $\mathcal{P}_5$ we can build the following arguments:

$\langle 2, \sim a \rangle$: $\langle \{\sim a \prec y, y \prec x, x \prec z\}, \sim a \rangle$
$\langle 3, \sim a \rangle$: $\langle \{\sim a \prec z\}, \sim a \rangle$
$\langle 4, a \rangle$: $\langle \{a \prec w, w \prec y, y \prec x\}, a \rangle$
$\langle 5, \sim a \rangle$: $\langle \{\sim a \prec t\}, \sim a \rangle$

Consider that $\mathcal{P}_5$ is revised by the argument $\langle 1, a \rangle$: $\langle \{a \prec x, x \prec z\}, a \rangle$. From now on, we abstract away both the argument preference criterion and dialectical line acceptability, so we will provide the attacks between arguments with no further discussion: arguments 2 and 3 attack 1, argument 4 attacks 3, and 5 attacks 4. Then, assume that the hypothetical D-tree on the right is built from $\mathcal{P}'_5 = (\Pi_5, \Delta_5 \cup 1)$. Here the argument selection function $\gamma^{\omega}_{\mathcal{P}}$ selects the arguments 2 and 3 to be cut off, while the argument incision function $\sigma^{\omega}_{\mathcal{P}}$ applied over them could be any subset.



To make an argument disappear, an incision over it must be performed. However, that incision might have a *collateral effect* and make another argument/s from the tree disappear. That is, the rules being cut off from an incised argument might belong to more arguments in the tree, and then the impact on the tree structure would be greater.

Definition 11 (Collateral Incision) *Let $\sigma^{\omega}_{\mathcal{P}}(\Psi)$ be an incision and $\mathcal{B}$ be any argument in the tree. If $\sigma^{\omega}_{\mathcal{P}}(\Psi) \cap \mathcal{B} \neq \emptyset$ holds, then $\sigma^{\omega}_{\mathcal{P}}(\Psi) \cap \mathcal{B}$ is called a* ***collateral incision*** *over $\mathcal{B}$.*

The argument incision function should be applied to the portion of the argument that does not belong to the root argument, *i.e.*, *it should avoid any collateral incision over the root argument.* The motivation of this property is that if a rule belonging to the root argument were to be cut off, this argument would no longer hold, turning impossible to warrant its conclusion. Therefore, the following property is proposed for an argument incision function "$\sigma^{\omega}_{\mathcal{P}}$":

(Root-Preservation) $\sigma^{\omega}_{\mathcal{P}}(\Psi_i) \cap \mathcal{A} = \emptyset$, where $\gamma^{\omega}_{\mathcal{P}}(\lambda_i) = \Psi_i$

Example 6 *Let us consider program $\mathcal{P}'_5$ from Ex. 5. Here, the incision over argument 2 could be any subset that does not contain the defeasible rule $x \prec\!\!- z$ (which belongs to the root argument 1), that is, any subset of $\{\sim a \prec\!\!- y, y \prec\!\!- x\}$. For instance, $\sigma^{\omega}_{\mathcal{P}}(2) = \{\sim a \prec\!\!- y\}$. The incision over argument 3, however, must be the single rule it contains: $\sigma^{\omega}_{\mathcal{P}}(3) = \{\sim a \prec\!\!- z\}$. Therefore we have that $\sigma^{\omega}_{\mathcal{P}}$ satisfies root-preservation.*

Remark 1 *Since arguments are minimal, given an argument $\langle \mathcal{B}, \beta \rangle$, it is clear that there is no defeasible derivation for β from $\Pi \cup (\mathcal{B} \setminus \sigma^{\omega}_{\mathcal{P}}(\mathcal{B}))$.*

When a collateral incision arises, some side effects may occur compromising the objective of the revision (*i.e.*, the root argument might end up defeated). This may happen in case a collateral incision affects a supporting argument in a dialectical line which originally had a defeated defeater (for the root), thus yielding it undefeated. This situation is captured by the following remark.

Remark 2 *The marking of a D-tree is considered dynamic, this is, it may change by a collateral effect of the applied incisions. Thereafter, if the status of a dialectical line has changed (now having an undefeated defeater for the root), then it should be further affected by an incision function.*

Definition 12 (Root-Preserving Argument Incision Function) *An argument incision function "$\sigma^{\omega}_{\mathcal{P}}$" determined by an argument selection function "$\gamma^{\omega}_{\mathcal{P}}$" is called **root-preserving argument incision function** if it verifies **root-preservation**.*

Now that both the selection and the incision function are defined, the WP Argument Revision operation can be formally defined.

Definition 13 (WP Argument Revision) *Let $\mathcal{P}$ be a program such that $\mathcal{P} = (\Pi, \Delta)$. A revision operation of $\mathcal{P}$ by an argument $\langle \mathcal{A}, \alpha \rangle$, namely $\mathcal{P} *^{\omega}_{\mathcal{P}} \langle \mathcal{A}, \alpha \rangle$, is defined by means of a **root-preserving argument incision function** "$\sigma^{\omega}_{\mathcal{P}}$" as follows:*

$$\mathcal{P} *^{\omega}_{\mathcal{P}} \langle \mathcal{A}, \alpha \rangle = (\Pi, \mathcal{A} \cup \Delta \setminus \bigcup_i (\sigma^{\omega}_{\mathcal{P}}(\Psi_i)))$$

Theorem 1 *Let $\mathcal{P}_R = \mathcal{P} *^{\omega}_{\mathcal{P}} \langle \mathcal{A}, \alpha \rangle$ be a revised defeasible logic program by an argument $\langle \mathcal{A}, \alpha \rangle$, then α is warranted from $\mathcal{P}_R$.*

Example 7 *From Ex. 6 we have the incisions $\sigma^{\omega}_{\mathcal{P}}(2) = \{\sim a \prec\!\!- y\}$ and $\sigma^{\omega}_{\mathcal{P}}(3) = \{\sim a \prec\!\!- z\}$. Then, from the revision $\mathcal{P}_5 *^{\omega}_{\mathcal{P}} \langle \{a \prec\!\!- x, x \prec\!\!- z\}, a \rangle$ made in Ex. 5 we have:*

$$\mathcal{P}_{5R} = (\Pi_5, \Delta_5 \cup \{a \prec\!\!- x, x \prec\!\!- z\} \setminus \{\sim a \prec\!\!- y, \sim a \prec\!\!- z\}) = \left(\left\{ \begin{array}{l} t, \\ z \end{array} \right\}, \left\{ \begin{array}{l} a \prec\!\!- x,\ y \prec\!\!- x,\ x \prec\!\!- z, \\ a \prec\!\!- w,\ w \prec\!\!- y,\ \sim a \prec\!\!- t \end{array} \right\} \right)$$

From $\mathcal{P}_{5R}$ literal a is warranted, since the D-tree is just the root of the one depicted in Ex. 5.

4.2 WP Argument Revision Considering Extended Arguments

Revising a program (Π, Δ) by an argument $\langle \mathcal{A}, \alpha \rangle$ assuming that $\mathcal{A} \cup \Pi$ derives α might be too restrictive. Then, from the operator explained in the last section, we can consider a variation of it that revises a program by an *extended argument*. These arguments will contain strict rules and facts, besides defeasible rules. This characteristic gives them the possibility of being *self-contained*, in the sense that they derive a conclusion just by themselves. However, extended arguments bring about a main drawback: consistency checking, *i.e.*, when a program is revised by an extended argument, the join of their sets of strict rules must be non-contradictory. Thereafter, this join can be defined following several policies, *i.e.*, deleting rules in contradiction, turn them into defeasible rules, *etc*. Moreover, this policy can be applied either over the strict rules of the program, over the strict rules of the argument, or both. Since we are defining a prioritized argument revision operator, we are going to keep the first option: only strict rules of the program will be affected. Next, we formally define the notion of extended argument.

Definition 14 (Extended Argument) *Given a Π set of strict rules and a set Δ of defeasible rules, a pair $\langle (\Pi, \Delta), \alpha \rangle$ is an extended argument structure for a literal α, if there is a minimal defeasible derivation for α from $\Pi \cup \Delta$, and $\Pi \cup \Delta$ is non-contradictory.*

Every concept from the DELP theory (such as (marked) dialectical trees) can be translated by replacing the classical notion of argument. A redefinition of these concepts is not to be done in this article, due to space reasons and because they are very similar to the original ones.

An extended-argument revision operation is noted as $\mathcal{P} \circledast^{\omega}_{\mathcal{P}} \langle (\Pi', \Delta'), \alpha \rangle$, where $\mathcal{P} = (\Pi, \Delta)$. Then, we should consistently join both strict sets Π and Π', by means of a *prioritized* multiple revision operator, namely "$\diamond_{\mathcal{P}}$", consequently defined as:

Definition 15 (Prioritized Multiple Revision) *Let "σ" be an incision function (Def. 4), K and A be two sets of sentences, such that K is consistent and A finite, and let $K^{\perp\!\!\!\perp}\beta$ be the set of β-kernels (Def. 3), such that $A \vdash \sim\beta$. The **Prioritized** Kernel Revision by a Set of Sentences operator "$\diamond_{\mathcal{P}}$" is defined as follows:*

$$K \diamond_{\mathcal{P}} A = (K \backslash \sigma(K^{\perp\!\!\!\perp}\beta)) \cup A$$

Note that the definition of "$\diamond_{\mathcal{P}}$" is inspired by the theory proposed in [5] and mostly by the definition of its non-prioritized version "$\diamond_{\mathcal{N}}$". The following properties are verified.

Proposition 1 *Given a **multiple prioritized revision** operation $K \diamond_{\mathcal{P}} A$, the following properties hold:*

1. $A \subseteq K \diamond_{\mathcal{P}} A$
2. $K \not\subseteq K \diamond_{\mathcal{P}} A$ *iff* $K \cup A \vdash \bot$

Therefore, as part of the definition of a *Warrant-Prioritized Extended-Argument Revision Operation* $\mathcal{P} \circledast^{\omega}_{\mathcal{P}} \langle (\Pi', \Delta'), \alpha \rangle$ (where $\mathcal{P} = (\Pi, \Delta)$), we should achieve the consistent joint of both strict sets of rules Π and Π', such that $\Pi \diamond_{\mathcal{P}} \Pi'$. Afterwards, the sets of defeasible rules Δ and Δ' should be joined. This may be easily achieved since there is no need to preserve consistence by a set of defeasible rules, then a preliminary version might be just $\Delta \cup \Delta'$. But furthermore, in order to preserve beliefs, a slight modification is proposed by adopting the policy of "*weakening*" the erased strict rules ρ, selected from Π by an incision function σ. Finally, while $\rho \in \sigma(\Pi^{\perp\!\!\!\perp}\beta)$ or equivalently $\rho \in \Pi \backslash \Pi \diamond_{\mathcal{P}} \Pi'$, the referred "*weakening*" is performed by means of a function δ such that $\delta(\rho)$ is the defeasible version of the strict rule ρ. This idea is originally exposed in [5], where a first approach

of revision in argumentative systems is given, and also in [14], where at one stage of the architecture two DELP-programs have to be combined.

Supposing that the operator "$\circledast^{\omega}_{\mathcal{P}}$" would define a new program $\mathcal{P}_R = (\Pi \diamond_{\mathcal{P}} \Pi', \Delta \cup \delta(\Pi \backslash \Pi \diamond_{\mathcal{P}} \Pi') \cup \Delta')$, it could not be possible to ensure that α is warranted from $\mathcal{P}_R$. Therefore, in order to achieve warrant for α, we propose to define the operator "$\circledast^{\omega}_{\mathcal{P}}$" by means of the operator "$*^{\omega}_{\mathcal{P}}$" previously defined, as follows:

Definition 16 (WP Extended-Argument Revision) *A Warrant-Prioritized Extended-Argument Revision Operator "$\circledast^{\omega}_{\mathcal{P}}$" is defined in terms of the operator "$\diamond_{\mathcal{P}}$" and the WP Argument Revision operator "$*^{\omega}_{\mathcal{P}}$" as follows:*

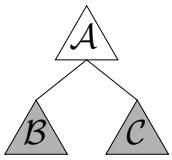
$$(\Pi, \Delta) \circledast^{\omega}_{\mathcal{P}} \langle (\Pi', \Delta'), \alpha \rangle = (\Pi \diamond_{\mathcal{P}} \Pi', \Delta \cup \delta(\Pi \backslash \Pi \diamond_{\mathcal{P}} \Pi')) *^{\omega}_{\mathcal{P}} \langle \Delta', \alpha \rangle$$

Example 8 *Consider the extended argument $\mathcal{A} = \langle (\{x \leftarrow t, t\}, \{a \prec\!\!\!- x\}), a \rangle$ and the* DELP*-program $\mathcal{P}_8 = (\{\sim x \leftarrow w, w \leftarrow z, z, y\}, \{\sim a \prec\!\!\!- y\})$. When joining $\mathcal{A}$ with $\mathcal{P}_8$, we have that there are strict derivations for both x (from the strict part of $\mathcal{A}$) and $\sim x$ (from the strict part of $\mathcal{P}_8$). Therefore, we apply the function δ to at least one strict rule from $\mathcal{P}_8$ that is involved in the derivation of $\sim x$; for instance, $\delta(w \leftarrow z) = w \prec\!\!\!- z$. Now we have:*

$$\mathcal{P}_8 \circledast^{\omega}_{\mathcal{P}} \mathcal{A} = (\{\sim x \leftarrow w, z, y, x \leftarrow t, t\}, \{\sim a \prec\!\!\!- y, w \prec\!\!\!- z\}) *^{\omega}_{\mathcal{P}} \langle \{a \prec\!\!\!- x\}, a \rangle$$

From this DELP*-program we can build the following extended arguments and dialectical tree:*

$\langle \mathcal{A}, a \rangle : \langle (\{x \leftarrow t, t\}, \{a \prec\!\!\!- x\}), a \rangle$
$\langle \mathcal{B}, \sim a \rangle : \langle (\{y\}, \{\sim a \prec\!\!\!- y\}), \sim a \rangle$
$\langle \mathcal{C}, \sim x \rangle : \langle (\{\sim x \leftarrow w, z\}, \{w \prec\!\!\!- z\}), \sim x \rangle$



Finally, as explained in the previous section, incisions over arguments $\mathcal{B}$ and $\mathcal{C}$ have to be made in order to turn $\mathcal{A}$ into an undefeated argument.

Remark 3 *Revising a program $\mathcal{P}$ by an extended argument $\langle (\Pi', \emptyset), \alpha \rangle$ is the case of an argument that, once introduced into $\mathcal{P}$, would have no argument against it (i.e., by definition, there would be no arguments for $\sim\alpha$), since α would have a strict derivation from Π'. Therefore, strict derivations for $\sim\alpha$ are to be weakened into defeasible derivations that, although they have no effect as arguments for $\sim\alpha$, they can be a part of other derivations that should not be "broken". This stresses the importance of not deleting conflicting strict rules, which can have undesirable collateral effects.*

5 Discussion, Conclusions & Future Work

In this work we have presented two different approaches for the WP Argument Revision operator: one considering regular arguments, and another, considering extended arguments. Both operators have a common theoretical basis, but the latter one has to resolve some extra issues.

The different versions of the *argument revision operator for defeasible logic programs* here proposed are based by most of the previous operator's definition. While *selections* may be related to the **partial-meet contractions** theory [2], *incisions* are inspired by **kernels contractions** [9]. Furthermore, the order established by a preference criterion on selections may be possibly related to **safe contractions** originally exposed in [3], and later on related to kernel contractions in [10]. Indeed,

an argument is a kind of kernel or minimal proof for a given consequence. These concepts are more deeply treated in [4], where the **Kernel Revision by a Set of Sentences** is proposed. Moreover, this operator constitutes part of the inspiration for the **argument revision operators** here exposed.

Our definition of the "$\diamond_{\mathcal{P}}$" operator ensures that there is no inconsistent intermediate epistemic state during the revision process. In [5], a non-prioritized revision operator over explanations "$\diamond_{\mathcal{N}}$" is introduced, which does generate an inconsistent intermediate epistemic state when the revision is performed. The latter operator justifies the non-prioritization by the assertion that there is no reason to accept new information blindly, discarding older beliefs without proper justification. We agree with this posture, but since we are defining an operator that has the objective of warranting the conclusion of the newly inserted external argument, we have to give priority to newer information. Let suppose the definition of a non-prioritized version of our operator by means of an operator "$\diamond_{\mathcal{N}}$":

$$(\Pi, \Delta) \circledast^{\omega}_{\mathcal{P}} \langle(\Pi', \Delta'), \alpha\rangle = (\Pi \diamond_{\mathcal{N}} \Pi', \Delta \cup \delta((\Pi \cup \Pi') \backslash (\Pi \diamond_{\mathcal{N}} \Pi')) *^{\omega}_{\mathcal{P}} \langle\Delta', \alpha\rangle$$

Note that the matter of de-prioritizing the incorporation of new information seems to be attained just to the join of the strict rules sets, while when we refer it as "warrant-prioritized", a reference to the priority of giving warrant to the new conclusion α is given. However, the definition of the non-prioritized operator has a major flaw: it does not ensure the warrant of α. An example will clarify this assertion:

Example 9 *Consider the extended argument* $\mathcal{A} = \langle(\Pi'_9, \{\}), a\rangle = \langle(\{a \leftarrow b, b\}, \{\}), a\rangle$ *and the* DELP*-program* $\mathcal{P}_9 = (\Pi_9, \{\}) = (\{\sim a \leftarrow c, c\}, \{\})$. *Here, if we prefer some older information over new one, we will have no argument for "*a*" from* $(\Pi_9 \diamond_{\mathcal{N}} \Pi'_9, \delta((\Pi_9 \cup \Pi'_9) \backslash (\Pi_9 \diamond_{\mathcal{N}} \Pi'_9)))$*, no matter if the rule* $(a \leftarrow b)$ *is turned into a defeasible rule (see Remark 3). Not having an argument for "*a*" makes impossible to have a warrant for it.*

In general, the theory we are defining cannot be trivially related to the basic concepts of belief revision. Regarding the basic postulates for a revision operator, as originally exposed in [2], a deep analysis is required. For example, the *success* postulate ($K * \alpha \vdash \alpha$) makes reference to a knowledge base K, which in our case is a DELP-program $\mathcal{P} = (\Pi, \Delta)$. For both of the argument revision operators here proposed, success is defined analogously, where the consequence notion is the warrant of the conclusion of the argument being added to $\mathcal{P}$. This statement is verified by Theorem 1. Another interesting postulate to be analyzed is *consistency*, which states that the outcome of a revision $K * \alpha$ must be consistent if α is non-contradictory. In our proposal, this postulate is treated in a trivial manner, since programs are revised by arguments, which are consistent by definition. Regarding extended arguments, a join between the strict parts of the program and the argument is performed to ensure consistency. DELP-programs are divided in two subsets of rules: Π and Δ, where only Π is required to be consistent, which is not modified by any of the argument revision operators we propose (because arguments do not introduce strict rules). Finally, the consistency postulate always hold for the two warrant-prioritized argument revision operators.

Besides exposing a complete list of the basic postulates and the respective axiomatic representations for each argument revision operator, future work also includes the definition of contraction/expansion of a DELP-program by an argument, and a detailed study towards the possibility of duality between the operators of contraction/expansion and revision in argumentation systems.

Optimality is not a property pursued in this work, since there are some cases in which incisions made in a lower level might compromise a smaller amount of program rules. Our major concern here is to define revision operators which just follow correctness regarding the objectives exposed above. That is, the main objective of this article is to present a first approach for revising defeasible

logic programs by an argument. Other possibilities besides warrant-prioritized argument revision are left unaddressed in this paper, as well as variations of the operators here defined. Some of these options are interesting, whereas others are trivial; however, the whole range of possibilities cannot be accounted on a single article. For instance, regarding the minimal change principle, at least two options arise: (1) we might introduce the notion of epistemic importance in an argument level, and thus would incise first those arguments that are less important wrt. the total epistemic order among them; (2) provided that D-trees are an important tool to understand the interrelation among arguments and their influence to the final answer, we might want to preserve the structure of the hypothetical D-tree; hence, incisions would be performed in its lowest levels.

References

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

[2] C. Alchourrón, P. Gärdenfors, and D. Makinson. *On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. The Journal of Symbolic Logic*, 50:510–530, 1985.

[3] C. Alchourrón and D. Makinson. On the logic of theory change: Safe contraction. *Studia Logica*, (44):405–422, 1985.

[4] M. Falappa. Teoría de Cambio de Creencias y sus Aplicaciones sobre Estados de Conocimiento. *Ph. D. Thesis*, June 1999.

[5] M. Falappa, G. Kern-Isberner, and G. R. Simari. Explanations, Belief Revision and Defeasible Reasoning. *Artificial Intelligence Journal*, 141(1-2):1–28, 2002.

[6] A. J. García and G. R. Simari. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.

[7] P. Gärdenfors. An Epistemic Approach to Conditionals. *American Philosophical Quarterly*, 18(3):203–211, 1981.

[8] P. Gärdenfors. Knowledge in Flux: Modelling the Dynamics of Epistemic States. *The MIT Press, Bradford Books, Cambridge, Massachusetts*, 1988.

[9] S. O. Hansson. Kernel Contraction. *The Journal of Symbolic Logic*, 59:845–859, 1994.

[10] S. O. Hansson. *A Textbook of Belief Dynamics: Theory Change and Database Updating. Springer.* 1999.

[11] I. Levi. Subjunctives, Dispositions, and Chances. *Synthèse*, 34:423–455, 1977.

[12] V. Lifschitz. Foundations of Logic Programs. In G. Brewka, editor, *Principles of Knowledge Representation*, pages 69–128. CSLI Pub., 1996.

[13] M. Moguillansky and M. Falappa. A Non-monotonic Description Logics Model for Merging Terminologies. *Revista Iberoamericana de Inteligencia Artificial (AEPIA), ISSN 1137-3601*, 2007. at press.

[14] N. D. Rotstein, A. J. García, and G. R. Simari. Reasoning From Desires to Intentions: A Dialectical Framework. *22nd. AAAI Conference on Artificial Intelligence (AAAI 2007), Vancouver, British Columbia, Canadá*, 2007. To appear.