Argument Theory Change Applied to Defeasible Logic Programming

Martín O. Moguillansky and **Nicolás D. Rotstein** and
Marcelo A. Falappa and **Alejandro J. García** and **Guillermo R. Simari**
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Artificial Intelligence Research and Development Laboratory (LIDIA)
Department of Computer Science and Engineering
Universidad Nacional del Sur (UNS), Bahía Blanca, Argentina
{mom,ndr,maf,ajg,grs}@cs.uns.edu.ar

Abstract

In this article we work on certain aspects of the belief change theory in order to make them suitable for argumentation systems. This approach is based on Defeasible Logic Programming as the argumentation formalism from which we ground the definitions. The objective of our proposal is to define an argument revision operator that inserts a new argument into a defeasible logic program in such a way that this argument ends up undefeated after the revision, thus warranting its conclusion. In order to ensure this warrant, the defeasible logic program has to be changed in concordance with a minimal change principle. Finally, we present an algorithm that implements the argument revision operation.

Introduction & Motivation

This work presents a first approach to introduce several concepts of belief revision within the area of argumentation systems. We are particularly focused on the revision of a knowledge base by an argument. To achieve this, we use Defeasible Logic Programming (DeLP) (García and Simari 2004) as the knowledge representation language, thus, knowledge bases will be represented as defeasible logic programs (DeLP-programs). The DeLP formalism is briefly described in the next section.

The main objective is to define an *argument revision operator* that ensures warrant of the conclusion of the (external) argument being added to a defeasible logic program. When we revise a program by an argument $\langle \mathcal{A}, \alpha \rangle$ (where $\mathcal{A}$ is an argument for α), the program resulting from the revision will be such that $\mathcal{A}$ is an undefeated argument and α is therefore warranted. In that sense, this operator will be *prioritized*. Thus, we refer to this operator as *warrant-prioritized argument revision operator* (WPA Revision Operator).

The main issue underlying warrant-prioritized argument revision (addressed in the third section of this paper) lies in the *selection* of arguments and the *incisions* that have to be made over them. An *argument selection criterion* will determine which arguments should not be present in order to ensure the inserted argument is undefeated. Once this selection is made, incisions (in the form of deletion of rules) will make those arguments "disappear"; but this process has to be done carefully, following some minimal change principle. In the fourth section, we present two examples of minimal change, thus defining the way the warrant-prioritized revision operator behaves. A general algorithm for argument revision is proposed in the closing part of this section. Finally, in the last section, related work is discussed, future work is proposed, and conclusions are drawn.

Elements of Defeasible Logic Programming

Defeasible Logic Programming (DeLP) combines results of Logic Programming and Defeasible Argumentation. The system is fully implemented and is available online (LIDIA 2007), and a brief explanation of its theory is included below. 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 "$\sim$". *Strict Rules* represent non-defeasible information noted as $\alpha \leftarrow \beta_1, \ldots, \beta_n$, where α is a ground literal and $\beta_{i>0}$ is a set of ground literals. *Defeasible Rules* represent tentative information noted as $\alpha \prec \beta_1, \ldots, \beta_n$, where α is a ground literal and $\beta_{i>0}$ is a set of ground literals.

When required, $\mathcal{P}$ will be denoted (Π, Δ) distinguishing the subset Π of facts and strict rules, and the subset Δ of defeasible rules (see Ex. 1). 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, that is, no pair of contradictory literals can be derived from Π. Strong negation can be used in the head of a rule, as well as in any literal in its body. In DeLP, literals can be derived from rules as in logic programming, being a *defeasible derivation* one that uses, at least, one defeasible rule.

Example 1 *Consider the* DeLP*-program* $\mathcal{P}_1 = (\Pi_1, \Delta_1)$*:*

$$\Pi_1 = \left\{ \begin{array}{c} t, z, \\ (p \leftarrow t) \end{array} \right\} \Delta_1 = \left\{ \begin{array}{c} (\sim a \prec y), (y \prec x), (x \prec z), \\ (y \prec p), (a \prec w), (w \prec y), \\ (\sim w \prec t), (\sim x \prec t), (x \prec p) \end{array} \right\}$$

From a program is possible to derive contradictory literals, *e.g.*, from (Π_1, Δ_1) of Ex. 1 it is possible to derive a and $\sim a$. DeLP incorporates a defeasible argumentation formalism for the treatment of contradictory knowledge. This

This work is partially supported by CONICET (PIP 5050), Universidad Nacional del Sur and ANPCyT.

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. In DeLP, an *argument* $\mathcal{A}$ is a minimal set of defeasible rules that, along with the set of strict rules and facts, is not contradictory and derives a certain conclusion α; this is noted as $\langle \mathcal{A}, \alpha \rangle$. As we will explain next, arguments supporting the answer for a given query are shown in a particular way using *dialectical trees*.

Example 2 *From $\mathcal{P}_1$ we can build the following arguments:*
$\langle \mathcal{B}_1, \sim a \rangle = \langle \{\sim a \prec y, y \prec x, x \prec z\}, \sim a \rangle$
$\langle \mathcal{B}_2, \sim a \rangle = \langle \{\sim a \prec y, y \prec p\}, \sim a \rangle$
$\langle \mathcal{B}_3, a \rangle = \langle \{a \prec w, w \prec y, y \prec p\}, a \rangle$
$\langle \mathcal{B}_4, \sim w \rangle = \langle \{\sim w \prec t\}, \sim w \rangle$
$\langle \mathcal{B}_5, \sim x \rangle = \langle \{\sim x \prec t\}, \sim x \rangle \quad \langle \mathcal{B}_6, x \rangle = \langle \{x \prec p\}, x \rangle$

A literal α is *warranted* if there exists a non-defeated argument $\mathcal{A}$ supporting α. 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 $\langle \mathcal{A}, \alpha \rangle$. An argument $\mathcal{A}_1$ is a counter-argument for $\mathcal{A}_2$ *iff* $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \Pi$ is contradictory. In DeLP, the comparison criterion is usually *generalized specificity* (Stolzenburg et al. 2003), but in the examples given in this paper we will abstract away from this criterion, since it introduces unnecessary complexity. Thus, the defeat relations between counter-arguments will be given explicitly by enumerating which argument defeats which other one. Since defeaters are arguments, there may exist defeaters for them, and defeaters for these defeaters, and so on. Thus, a sequence of arguments called *argumentation line* is constructed, where each argument defeats its predecessor. To avoid undesirable sequences, that may represent circular or fallacious argumentation lines, in DeLP an argumentation line has to be *acceptable*, that is, it has to be finite, an argument cannot appear twice, and supporting (resp., interfering) arguments have to be non-contradictory. From now on every argumentation line will be assumed acceptable.

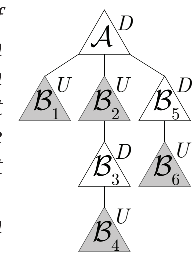
Example 3 *Consider the arguments from Ex. 2. For simplicity, we will provide the following defeat relation: $\mathcal{B}_3$ defeats $\mathcal{B}_2$, and $\mathcal{B}_4$ defeats $\mathcal{B}_3$. From these three arguments we can build the sequence $[\mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4]$, which 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. This tree is called *dialectical* because it represents an exhaustive dialectical analysis for the argument in its root. 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.

Given a literal α and an argument $\langle \mathcal{A}, \alpha \rangle$ from a program $\mathcal{P}$, to decide whether α is warranted, every node in the tree is recursively marked as "D" (*defeated*) or "U" (*undefeated*), obtaining a marked dialectical tree $\mathcal{T}_{\mathcal{P}}(\mathcal{A})$: (1) all leaves in $\mathcal{T}_{\mathcal{P}}(\mathcal{A})$ are marked as "U"s; and (2) let $\mathcal{B}$ be an inner node of $\mathcal{T}_{\mathcal{P}}(\mathcal{A})$, then $\mathcal{B}$ will be marked as "U" *iff* every child of $\mathcal{B}$ is marked as "D". Thus, the node $\mathcal{B}$ will be marked as "D" *iff* it has at least one child marked as "U".

Given an argument $\langle \mathcal{A}, \alpha \rangle$ obtained from $\mathcal{P}$, if the root of $\mathcal{T}_{\mathcal{P}}(\mathcal{A})$ is marked as "U", then we say that $\mathcal{T}_{\mathcal{P}}(\mathcal{A})$ *warrants* α and that α is *warranted* from $\mathcal{P}$. When no confusion arises, we will refer to $\mathcal{A}$ instead of α, saying that $\mathcal{A}$ is warranted. In this paper, an argument is depicted as a triangle; gray triangles will be undefeated arguments, whereas white triangles will depict defeated arguments. Therefore, marked dialectical trees will be represented as a tree of triangles, where edges denote the defeat relation. In the rest of the article we refer to marked dialectical trees just as "trees".

Example 4
From the DeLP*-program (Π_1, Δ_1) of Ex. 1 we can consider a new program $\mathcal{P}_4 = (\Pi_1, \Delta_1 \cup \{a \prec x\})$, from which we can build the additional argument $\langle \mathcal{A}, a \rangle = \langle \{(a \prec x), (x \prec z)\}, a \rangle$. The defeat relations are: $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{B}_5$ defeat $\mathcal{A}$, $\mathcal{B}_3$ defeats $\mathcal{B}_2$, $\mathcal{B}_4$ defeats $\mathcal{B}_3$, and $\mathcal{B}_6$ defeats $\mathcal{B}_5$. The tree $\mathcal{T}_{\mathcal{P}_4}(\mathcal{A})$ depicted on the right is the tree for $\mathcal{A}$ from $\mathcal{P}_4$.*



For simplicity, those arguments that can be built from $\mathcal{P}_4$ but do not appear in the tree $\mathcal{T}_{\mathcal{P}_4}(\mathcal{A})$ are assumed to be defeated by the corresponding arguments that do appear. For instance, there is an argument $\langle \{\sim a \prec y, y \prec x, x \prec p\}, \sim a \rangle$ that we assume is defeated by $\mathcal{A}$ and $\mathcal{B}_3$.

An Argument Revision Operator

Intuitively, a *Warrant-Prioritized Argument Revision Operator* (for short: WPA Revision Operator) revises a given program $\mathcal{P} = (\Pi, \Delta)$ by an external argument $\langle \mathcal{A}, \alpha \rangle$ ensuring $\mathcal{A}$ ends up being warranted from the program resulting from the revision –provided that $\mathcal{A} \cup \Pi$ has a defeasible derivation for α. This condition for deriving α relies on the fact that the set Π of strict rules and facts represents (in a way) the current state of the world. The 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.*, $\mathcal{A} \cup \Pi$ defeasibly derives α.

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: to ensure $\mathcal{A}$ to be an undefeated argument, and thus, α would be always warranted (at least because of $\mathcal{A}$).

For this matter, a *hypothetical dialectical tree* rooted on $\langle \mathcal{A}, \alpha \rangle$ from program $\mathcal{P}' = (\Pi, \Delta \cup \mathcal{A})$ is built. The tree is deemed as "hypothetical" because it does not belong to the original program $\mathcal{P}$ and (possibly) neither to the program P_R resulting from the revision. That is, it belongs to the program $\mathcal{P}'$, which represents an intermediate state. This state is not final and consequently the tree is hypothetical, since it would suffer some modifications in order to warrant α. Finally, since the tree is modified through incisions over interference arguments we need to be able to identify them.

Definition 1 (Set of Interference (Supporting) Arguments) *Let $\lambda = [\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n]$ be an argumentation line, then the* ***set of interference*** *(resp.,* ***supporting****)* ***arguments*** *λ^- (resp., λ^+) of λ is the set containing all the arguments placed on even (resp., odd) positions in λ.*

Remark 1 *Just for simplicity, from now on we use $\mathcal{B}_i^+$ to mean that $\mathcal{B} \in \lambda_i^+$, and similarly $\mathcal{B}_i^-$ wrt. $\mathcal{B} \in \lambda_i^-$.*

Once an argument of a line λ is incised (*i.e.*, it disappears), it is necessary to identify the part of λ that remains as a part of the tree, either within another line or as a separate line.

Definition 2 (Upper Segment) *Let λ be the argumentation line $[\mathcal{B}_1, \ldots, \mathcal{B}_j, \ldots, \mathcal{B}_n]$. The* ***upper segment*** *of λ wrt. $\mathcal{B}_j$ is $\lambda^\uparrow(\mathcal{B}_j) = [\mathcal{B}_1, \ldots, \mathcal{B}_{j-1}]$, and $\lambda^\uparrow(\mathcal{B}_1)$ does not exist.*

In a dialectical tree, we need to characterize the kind of argumentation lines that actually affect the "defeated" status of the root argument. Next, we define the notion of *attacking lines*, which are the lines in a tree over which the argument selection and then the argument incision are going to be applied. Intuitively, the set of attacking lines is the minimal subset of lines from a tree such that, without them, the tree would warrant its root argument (see Lemma 1).

The marking of an argumentation line will not be considered individually, but in concordance with the context provided by the tree it belongs to. For instance, in Ex. 4 the line $[\mathcal{A}, \mathcal{B}_5, \mathcal{B}_6]$ does not have the marking sequence UDU but the marking DDU, since $\mathcal{B}_1$ (from line $[\mathcal{A}, \mathcal{B}_1]$) is an undefeated defeater for $\mathcal{A}$, which is therefore defeated (marked as D). In general, the marking sequence of any argumentation line can be associated to a regular expression.

Definition 3 (Attacking lines) *Let λ be an argumentation line. We call λ an* ***attacking line*** *iff its marking sequence corresponds to the regular expression $(DU)^+$.*

Example 5 *From the tree of Ex. 4 we have two attacking lines: $[\mathcal{A}, \mathcal{B}_1]$ and $[\mathcal{A}, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4]$. Note that a possible tree $\mathcal{T}_{\mathcal{P}'_4}(\mathcal{A})$ containing just the line $[\mathcal{A}, \mathcal{B}_5, \mathcal{B}_6]$ would warrant $\mathcal{A}$, since its marking sequence is UDU. The addition of any attacking line to $\mathcal{T}_{\mathcal{P}'_4}(\mathcal{A})$ implies $\mathcal{A}$ to be defeated.*

The argument theory change proposed here follows a minimal change principle. In this sense, a particular change operation could avoid the complete erasure of attacking lines, and a portion of them could still appear in the resulting tree.

Definition 4 (Types of Argumentation Lines) *There are two types of argumentation lines regarding their marking:*

1. *Warranting Lines: $U(D^+U)^*$*
2. *Non-warranting Lines: $(D^+U)^+$*

The notation ***W lines*** *identifies the warranting lines in general; whereas the non-warranting are subdivided in* ***attacking lines*** *($(DU)^+$) and* ***D-rep lines*** *(those that have a repetition of Ds in at least one place in the sequence).*

Some W lines also have a repetition of Ds in their sequence, but we do not distinguish this kind of lines because they do not require a separate analysis –they do not threat the warrant of the root argument. Despite being non-warranting, D-rep lines are not "responsible" for the mark D of the root argument: it is such due to the D-rep lines sharing their beginning with an attacking line.

Proposition 1 [1] *If λ_D is a D-rep line, then there exist an associated attacking line λ_A and two arguments $\mathcal{B}_D \in \lambda_D$, $\mathcal{B}_A \in \lambda_A$ such that $\lambda_D^\uparrow(\mathcal{B}_D) = \lambda_A^\uparrow(\mathcal{B}_A)$.*

Given a tree, for every attacking line λ, an interference argument is selected over λ^- on behalf of an *argument selection criterion* "$\prec_\gamma$" by the use of an *argument selection function*, namely γ^ω.

Definition 5 (Argument Selection Criterion "$\prec_\gamma$") *Let Γ be a set of arguments, "$\prec_\gamma$" is an* ***argument selection criterion*** *iff Γ is a totally ordered set wrt. the operator "$\prec_\gamma$".*

In order to warrant the argument in the root of the hypothetical tree, we need an argument selection function that returns which argument should be "erased" from a given line. This should not be performed randomly, but following some policy, namely the *minimal change principle*. Therefore, the deletion of the selected argument will conform this principle, ensuring a minimal "amount" of change is provoked.

The argument selection function is tied to the argument selection criterion "$\prec_\gamma$", that in turn is guided by the minimal change principle, which organizes a set of arguments according to the order "$\prec_\gamma$". This set comes from an attacking line, over which we select an argument to put away.

Definition 6 (Argument Selection Function "γ^ω") *An* ***argument selection function*** *"γ^ω" is applied over every attacking line λ_i. Therefore, a selection $\gamma^\omega(\lambda_i) = \mathcal{B}_i^-$ is determined by the selection criterion "$\prec_\gamma$" and it is called Ψ_i.*

Proposition 2 *The upper segment of a selected argument in an attacking line is a non-attacking line.*
Proof sketch: An attacking line $\lambda = [\mathcal{A}_1, \ldots, \mathcal{A}_j, \ldots, \mathcal{A}_n]$ has a marking sequence $(DU)^+$. Assuming that $\mathcal{A}_j \in \lambda^-$ is selected, if its upper segment $\lambda^\uparrow(\mathcal{A}_j) = [\mathcal{A}_1, \ldots, \mathcal{A}_{j-1}]$ were an acceptable argumentation line (after the incision of $\mathcal{A}_j$), then $\lambda^\uparrow(\mathcal{A}_j)$ would be either a W line or a D-rep line. Thus, none of them is an attacking line.

After an argument is selected, it is desirable that the subsequent incision over this argument does not affect other arguments in the tree. For this matter, the following property is proposed for an argument selection function:

$$\textbf{(Cautiousness)}\ \gamma^\omega(\lambda_i) \setminus \bigcup \mathcal{B} \neq \emptyset, \text{ for every } \mathcal{B} \in \mathcal{T}_{\mathcal{P}}(\mathcal{A}), \mathcal{B} \neq \gamma^\omega(\lambda_i)$$

Here, two possibilities arise: either the whole selected interference argument or just a portion of it does not "collide" with any supporting argument of the tree.

Definition 7 (Cautious and Non-Cautious Selections) *Given an argument selection function γ^ω and an argumentation line λ, then $\gamma^\omega(\lambda)$ is identified as a* ***cautious***

[1] Some proofs in this article were omitted due to space reasons.

selection Ψ *iff* γ^ω *verifies* ***cautiousness****. Otherwise,* $\gamma^\omega(\lambda)$ *is identified as a* ***non-cautious selection*** Ψ*.*

Example 6 *From the tree of Ex. 4, it can be seen that the only selection in the attacking line* $[\mathcal{A}, \mathcal{B}_1]$ *is* $\mathcal{B}_1$*, whereas for the attacking line* $[\mathcal{A}, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4]$*, the selection function could return either* $\mathcal{B}_2$ *or* $\mathcal{B}_4$*. Regarding the selection of* $\mathcal{B}_4$*, it satisfies cautiousness because it has no intersection with any other argument. In contrast, the selection of* $\mathcal{B}_2$ *would be non-cautious, since its two rules* $\sim a \prec y$ *and* $y \prec p$ *belong to* $\mathcal{B}_1$ *and* $\mathcal{B}_3$*, respectively. Finally, considering* $\mathcal{B}_1$ *in the other attacking line, we have that* $\mathcal{B}_1 \cap \mathcal{A} = \{x \prec z\}$ *and* $\mathcal{B}_1 \cap \mathcal{B}_2 = \{\sim a \prec y\}$*. However, the remaining portion of* $\mathcal{B}_1$ *is non-empty, that is* $\mathcal{B}_1 \setminus \bigcup(\mathcal{A}, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_6) = \{y \prec x\}$*; hence, the selection of* $\mathcal{B}_1$ *verifies cautiousness.*

After the argument selection, an *argument incision function* σ^ω is applied to the selected argument Ψ_i, identifying the non-empty set of cut-off defeasible beliefs. Once the cut over an interference argument is made, the attacking line it belonged to turns into a non-attacking line (see Proposition 2). Analogously to the case of the selection function and the need of a selection criterion, we define an incision criterion to guide the incision function, which should be also related to the minimal change principle.

Definition 8 (Argument Incision Criterion "$\prec_\sigma$") *Let* Γ *be a set of defeasible rules, "*$\prec_\sigma$*" is an* ***incision selection criterion*** *iff* Γ *is a totally ordered set wrt. the operator "*$\prec_\sigma$*".*

Definition 9 (Argument Incision Function "σ^ω") *Let* Ψ *be an argument determined by the selection function "*γ^ω*". Then a function "*σ^ω*" is an* ***argument incision function*** *iff it is determined by the incision criterion "*$\prec_\sigma$*" and verifies* $\emptyset \subset \sigma^\omega(\Psi) \subseteq \Psi$*.*

Remark 2 *Since arguments are minimal, given an argument* $\langle \mathcal{B}, \beta \rangle$ *it is clear that there is no defeasible derivation for* β *from* $(\mathcal{B} \setminus \sigma^\omega(\mathcal{B})) \cup \Pi$*.*

To make an argument disappear, an incision over it must be applied. However, that incision might have a *collateral effect* and make another argument/s from the tree also disappear. We call this effect *collateral incision*. When this occurs over more than one argument in the same line, we will be interested in the uppermost one, since its disappearance will make the lower ones also disappear.

Definition 10 (Collateral Incision) *Let* $\sigma^\omega(\Psi)$ *be an incision and* $\mathcal{B} \in \lambda$*, an argument. A* ***collateral incision*** *over* $\mathcal{B}$ *is defined as* $\sigma^\omega(\Psi) \cap \mathcal{B} \neq \emptyset$*. If* $\sigma^\omega(\Psi) \cap \mathcal{C} = \emptyset$ *for every* $\mathcal{C} \in \lambda^\uparrow(\mathcal{B})$*, we have that* $\sigma^\omega(\Psi)^{(\mathcal{B})} = \sigma^\omega(\Psi) \cap \mathcal{B}$ *is the* ***uppermost collateral incision*** *over* λ*.*

In what follows, we provide some tools to handle collateral incisions by proposing some properties for an incision function "σ^ω". The following section works on this topic by describing a way to take advantage of these properties.

(Strict-Preservation) $\sigma^\omega(\Psi)^{(\mathcal{B})} = \emptyset$,
for any $\mathcal{B}$ in any λ in $\mathcal{T}_{\mathcal{P}}(\mathcal{A})$

Although strict-preservation might be a desirable property, it is not always possible to verify it. Thus, when a collateral incision is unavoidable, some side effects may occur compromising the goal of the revision (*i.e.*, the root argument might end up defeated). To avoid this, we have to restrict the selection performed over the argumentation line where the collateral incision arises: the collaterally incised argument should not be in the upper segment of the argument selected, since the selection would have no effect.

(Preservation) *If* $\sigma^\omega(\Psi_i)^{(\mathcal{B}_j)} \neq \emptyset$ *then*
(exists $\lambda_j^\uparrow(\mathcal{B}_j)$) *and* ($\Psi_j \in \lambda_j^\uparrow(\mathcal{B}_j)$ *iff* $\lambda_j^\uparrow(\mathcal{B}_j)$ is att. line),
for any $\mathcal{B}_j$

This principle is illustrated in Fig. 1, in which the selection is labeled with a minus symbol. When an incision $\sigma^\omega(\Psi_i)$ in the i^{th} dialectical line (the left branch in Fig. 1) results in an (uppermost) collateral incision $\sigma^\omega(\Psi_i)^{(\mathcal{B}_j)}$ over argument $\mathcal{B}_j$ in the j^{th} dialectical line (right branch), it must be ensured that the selection Ψ_j in the j^{th} line is performed over the upper segment $\lambda_j^\uparrow(\mathcal{B}_j)$. This selection is only performed if $\lambda_j^\uparrow(\mathcal{B}_j)$ is an attacking line. Finally, note that if $\mathcal{B}_j$ were the root node, then there would be no upper segment for it. In the case of the antecedent of the principle being false; that is, when there is no collateral incision over any argument $\mathcal{B}_j$ in any j^{th} line, the validity of the preservation principle is not threatened. Note that this particular case subsumes strict-preservation.

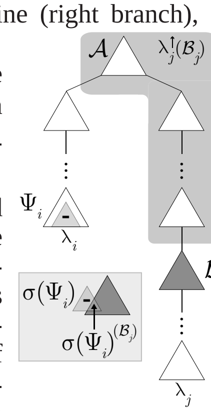


Figure 1 Preservation

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 clear: 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 incision function "σ^ω":

(Root-Preservation) $\sigma^\omega(\Psi)^{(\mathcal{A})} = \emptyset$

Root-preservation is a particular case of **strict-preservation**, where the argument $\mathcal{B}_j$ is the root argument $\mathcal{A}$. Regarding **preservation**, a collateral incision over $\mathcal{A}$ would not be possible, since $\lambda^\uparrow(\mathcal{A})$ does not exist, and the consequent of this principle would be false, which means that the antecedent should also be false in order for the principle to hold. This is so when root-preservation is satisfied. The interrelation among the three preservation principles is shown by the following proposition.

Proposition 3 *If "*σ^ω*" verifies* ***strict-preservation*** *then it also verifies* ***root-preservation*** *(****strict-preservation*** *implies* ***root-preservation****). Similarly,* ***preservation*** *implies* ***root-preservation*** *and* ***strict-preservation*** *implies* ***preservation****.*

Remark 3 ***Collateral incisions*** *may occur in the same argumentation line than the incision* $\sigma^\omega(\Psi)$*, but they cannot occur in the upper segment of the selected argument* Ψ*, since that would not verify* ***preservation****.*

Proposition 4 *A selection Ψ is **cautious** iff there exists an incision $\sigma^{\omega}(\Psi)$ verifying **strict-preservation**.*

Definition 11 (Warranting Incision Function) *An argument incision function "σ^{ω}" verifying preservation is said to be a **warranting incision function**.*

Finally, the WPA Revision is formally defined as:

Definition 12 (WPA Revision) *Let $\mathcal{P} = (\Pi, \Delta)$ be a* DELP*-program. A revision operator of $\mathcal{P}$ by an argument $\langle \mathcal{A}, \alpha \rangle$, namely $\mathcal{P} *^{\omega} \langle \mathcal{A}, \alpha \rangle$, is defined by means of a **warranting incision function** "σ^{ω}" as follows:*

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

Lemma 1 *A dialectical tree containing no attacking lines has its root marked as undefeated.*

Proof sketch: A restatement of this lemma is: if the tree contains no attacking lines, then it only contains W lines. The tree cannot contain non-warranting lines because they are either attacking or D-rep. The former violates the hypothesis, and the latter cannot exist without an associated attacking line, as stated by Proposition 1.

Theorem 1 *Let $\mathcal{P}$ be a* DELP*-program, "$*^{\omega}$", a WPA Revision Operator, and $\langle \mathcal{A}, \alpha \rangle$, an argument. Then α is warranted from $\mathcal{P} *^{\omega} \langle \mathcal{A}, \alpha \rangle$.*

Proof sketch: For each attacking line in the hypothetical tree rooted in $\langle \mathcal{A}, \alpha \rangle$, the selection gives an interference argument, the incision there makes that argument disappear, and the remaining upper segment (by Proposition 2) is a non-attacking line. Therefore, the tree no longer contains attacking lines and (by Lemma 1) it warrants α.

Two Minimal Change Principles

In this section we propose two minimal change principles: preserving program rules and preserving the hypothetical tree structure. The definition of a new minimal change principle depends on the explicit definition of the criteria for selections "$\prec_{\gamma}$" and incisions "$\prec_{\sigma}$". Therefore, for each principle we are going to define the corresponding selection criterion. The incision criterion "$\prec_{\sigma}$" will be common to both principles, and would assume, for instance, some kind of epistemic entrenchment over rules of each argument. Finally, one restriction over the relation of selections and incisions will be considered for each proposed criterion. These restrictions use the tools provided, like cautiousness and the three preservation principles.

(1) Preserving Program Rules. In general, selecting directly some defeaters for the root argument ensures a minimal deletion of defeasible rules from the DELP-program. This is so because the deletion of a root's defeater eliminates a whole branch. Trying to achieve the same result by deleting rules from "lower" arguments in the tree would affect a greater amount of arguments, due to the possibility of branching. In order for the root argument to be warranted, its defeaters have to be defeated. Therefore, we need to make incisions over those defeaters that are undefeated, *i.e.*, those that belong to attacking lines.

Rules-Preserving Selection Criterion. Given an attacking line $\lambda = [\mathcal{A}, \mathcal{B}_2, \ldots, \mathcal{B}_n]$, let $\Gamma = \lambda^{-} = \bigcup_k \{\mathcal{B}_k\}$, with k even, we have that $\mathcal{B}_i \prec_{\gamma} \mathcal{B}_j$, for $i < j$.

An interesting *restriction* regarding incisions is to seek for those that have a collateral incision with a selection in another line. Such an incision is desirable, since it would not only save a future incision, but would also prevent the deletion of extra rules.

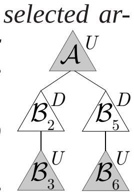
(2) Preserving Tree Structure. When trees are treated as an explanation for the DELP-answer given to a query (García et al. 2007; García, Rotstein, and Simari 2007), they are of upmost importance, since their structure is the main source of information. Provided that trees are an important tool to understand the interrelation among arguments and their influence to the final answer (and its trust), we will define a selection criterion that determines a revision operation making minimal changes in the structure of the hypothetical tree in order to render its root undefeated. Therefore, the selection criterion will be determined by the level of the argument in the argumentation line; the lower an argument is, the less is its impact in the structure of the tree, making the argument more suitable for selection. Hence, this criterion specifies the opposite order than the rules-preserving one, *i.e.*, "Given an attacking line [...] $\mathcal{B}_i \prec_{\gamma} \mathcal{B}_j$, for $i > j$."

In this case, an interesting restriction is to identify those strict-preserving incisions, that is, incisions that do not collide with any other argument in the tree. This would collaborate with the preservation of the tree structure.

Example 7 *Consider the program $\mathcal{P}_1$ being revised by the argument $\mathcal{A}$ and the corresponding hypothetical tree of Ex. 4. The criterion dedicated to **preserve program rules** would select arguments $\mathcal{B}_1$ and $\mathcal{B}_2$. From Ex. 6 we know that there is a way of incising $\mathcal{B}_1$ while collaterally incising $\mathcal{B}_2$, which is $\sigma^{\omega}(\mathcal{B}_1) = \{\sim a \mathrel{-\!\!\prec} y\}$. Therefore, the rules-preserving revised program would lose just one rule:*

$$\mathcal{P}^1_R = (\Pi_1, \Delta_1 \cup \{\mathcal{A}\} \setminus \{\sim a \mathrel{-\!\!\prec} y\})$$

*Following the **tree-preserving** minimal change principle, lower selections are considered first, thus the selected arguments are $\mathcal{B}_1$ and $\mathcal{B}_4$. Now the incision over $\mathcal{B}_1$ will avoid collateral incisions, i.e., will be strict-preserving; hence, $\sigma^{\omega}(\mathcal{B}_1) = \{y \mathrel{-\!\!\prec} x\}$. Since $\mathcal{B}_4$ is a cautious selection (see Ex. 6) and has one rule, the only possible incision is: $\sigma^{\omega}(\mathcal{B}_4) = \{\sim w \mathrel{-\!\!\prec} t\}$. Finally, the resulting tree is depicted on the right and its corresponding program is:*



$$\mathcal{P}^2_R = (\Pi_1, \Delta_1 \cup \{\mathcal{A}\} \setminus \{(y \mathrel{-\!\!\prec} x), (\sim w \mathrel{-\!\!\prec} t)\})$$

An Algorithm for the Revision. The PROLOG-like Algorithm 1 for the argument revision considers a possible implementation of this operation[2]. Predicate $revise/3$ takes a program and an argument, performs the revision, and returns the revised program. In order to gather all the incisions performed over attacking lines, predicate $get\_incisions/1$ performs the computation of these incisions over each attacking line through predicate $do\_incision/1$, attempting to incise the best selection wrt. the adopted selection criterion. A sequence of facts $select/2$ ordered according to this criterion implements the selection function. If the best selection does not imply the best incision (according to the corre-

[2]Note that all the symbols are variables, *i.e.*, there are no atoms.

sponding predicate $restrictions/2$), a "second best" selection is considered by backtracking, and so on until the best combination of selection/incision is found. In case this does not happen, the "first best" selection is reconsidered, with no regard about the quality of the incision. If no restriction is verified, the incision is only guided by preservation and, as stated above, the best selection is reconsidered.

Algorithm 1 Argument Revision

$revise((\Pi, \Delta), \langle \mathcal{A}, \alpha \rangle, (\Pi, \Delta_R)) \leftarrow$
$\quad union(\Delta, \mathcal{A}, \Delta_{\mathcal{A}}),$
$\quad assert\_att\_lines((\Pi, \Delta_{\mathcal{A}})),$ `%facts attacking/1`
$\quad assert\_lines((\Pi, \Delta_{\mathcal{A}})),$ `%facts line/1`
$\quad get\_incisions(\Sigma), subtract(\Delta_{\mathcal{A}}, \Sigma, \Delta_R).$

$get\_incisions(\Sigma) \leftarrow$
$\quad findall(\sigma, do\_incision(\sigma), \Sigma).$

$do\_incision(\sigma) \leftarrow$
$\quad retract(attacking(\lambda)),$
$\quad (select(\Psi, \lambda), restrictions(\Psi, \sigma), !$
$\quad ;$
$\quad select(\Psi, \lambda), incise(\sigma, \Psi)).$

(1) $restrictions(\Psi_i, \sigma) \leftarrow$ `%preserving program rules`
$\quad attacking(\lambda_j), select(\Psi_j, \lambda_j), incise(\sigma, \Psi_i),$
$\quad intersect(\sigma, \Psi_j), retract(attacking(\lambda_j)).$

(2) $restrictions(\Psi_i, \sigma) \leftarrow$ `%preserving tree structure`
$\quad incise(\sigma, \Psi_i),$
$\quad not((line(\lambda_j), member(\mathcal{B}, \lambda_j), intersect(\sigma, \mathcal{B}))).$

Discussion, Conclusions & Future Work

The main objective of this article was to present a first approach to revise defeasible logic programs by an argument. An Argument Revision Operator was thus proposed, along with two minimal change principles: preservation of program rules and tree structure. Both principles throw different results when applied to the same program (see Ex. 7). Additional rationality principles and change operators are left unaddressed in this paper. For instance, regarding minimal change, a different approach would attempt to preserve the semantics of the program (*i.e.*, the set of warranted literals).

The WPA Revision Operator here proposed may be compared to some operations in the theory change. While *selections* in WPA Revision may be related to selections in the *partial-meet contractions* theory (Alchourrón, Gärdenfors, and Makinson 1985), *incisions* are inspired by *kernels contractions* (Hansson 1994). Furthermore, the order established by a preference criterion on selections may be possibly related to *safe contractions* as originally exposed in (Alchourrón and Makinson 1985), and later on related to kernel contractions in (Hansson 1999). Indeed, an argument is a kind of kernel or minimal proof for a given consequence. These concepts are more deeply treated in (Falappa, Kern-Isberner, and Simari 2002), where the *kernel revision by a set of sentences* is proposed. Moreover, this operator constitutes part of the inspiration for the *argument revision operator*.

However, 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 (Alchourrón, Gärdenfors, and Makinson 1985), a more detailed 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 $\mathcal{P} = (\Pi, \Delta)$, and the sentence α is generalized to a set of defeasible rules (*i.e.*, an argument). Thus, success can be defined analogously, where the notion of consequence is warrant. 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, and the only subset of a DeLP-program that is required to be consistent (*i.e.*, Π) is not modified by the argument revision. Finally, consistency would always hold for the WPA Revision Operator.

Future work includes a proposal of a set of rationality principles suggesting a set of basic postulates, along with their respective analysis and an axiomatic representation for the WPA Revision Operator. Besides, ongoing work comprehends the definition of new change operators, such as contraction/expansion of a DeLP-program by an argument, followed by a detailed study towards duality between contraction/expansion and revision in argumentation systems.

References

Alchourrón, C., and Makinson, D. 1985. On the Logic of Theory Change: Safe Contraction. *Studia Logica* (44):405–422.

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

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

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

García, A.; Chesñevar, C.; Rotstein, N.; and Simari, G. 2007. An Abstract Presentation of Dialectical Explanations in Defeasible Argumentation. In *ArgNMR'07*, 17–32.

García, A.; Rotstein, N.; and Simari, G. 2007. Dialectical Explanations in Defeasible Argumentation. In *ECSQARU'07*, 295–307.

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

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

LIDIA. 2007. DeLP Web Client and Visualization Tool. http://lidia.cs.uns.edu.ar/delp_client.

Stolzenburg, F.; García, A.; Chesñevar, C. I.; and Simari, G. R. 2003. Computing Generalized Specificity. *Journal of Non-Classical Logics* 13(1):87–113.