

# Preferential Defeasibility: Utility in Defeasible Logic Programming \*

Fernando A. Tohmé<sup>1</sup> and Guillermo R. Simari<sup>2</sup>

ftohme@criba.edu.ar grs@cs.uns.edu.ar

Artificial Intelligence Research and Development Laboratory

<sup>1</sup>Department of Economics and National Research Council (CONICET)

<sup>2</sup>Department of Computer Science and Engineering

Universidad Nacional del Sur

Av. Alem 1253, (8000) Bahía Blanca, Argentina

## Abstract

The development of Logic Programming and Defeasible Argumentation lead to Defeasible Logic Programming. Its core resides in the characterization of the warrant procedure. Defeasible Argumentation has provided a solid foundation over which the standard formalization of this procedure has been constructed. A key element in the warrant procedure is the criterion according to which two contradicting arguments are compared and eventually one of them deemed as defeating the other. The purely syntactic Specificity criterion has constituted the main choice in the design of the warrant procedure. Nevertheless, it seems unreasonable to limit the possibilities of comparison among arguments only to syntactic criteria.

The justification of the methods of Defeasible Argumentation are largely pragmatic. Therefore, it seems sensible to expand the set of comparison criteria to incorporate other pragmatic reasons for choosing one argument over another. Decision Theory is the natural choice to model decision-makers. Clearly, as a discipline, it has characterized and introduced formal models in all kinds of pragmatic criteria used in actual choice situations.

Here, we will present the framework of Preferential Defeasible Logic Programming. This framework extends the original comparison criteria of specificity redefining it by allowing different preferential values for activation sets. This extension leads to interesting results where the decision is taken considering not only specificity, but also the corresponding pragmatic relation of preferences.

## Introduction and Motivation

The development of *defeasible reasoning* in the last decades (Pollock 1987; Simari & Loui 1992; Nute 1994; Pollock 1995; Chesñevar, Maguitman, & Loui 2000), lead to the creation of an alternative form of declarative programming, Defeasible Logic Programming

(DeLP) (García 2000; García & Simari 2004). This formalism blends Logic Programming with Defeasible Argumentation, allowing the representation of tentative knowledge and leaving for the inference mechanism the task of finding the conclusions that the knowledge base warrants (Chesñevar *et al.* 2003).

DeLP inherits from Logic Programming (LP) the formal characterization of *programs* as sets of rules. The difference is that DeLP considers two kinds of rules. On one hand, *strict rules*, which are assumed to represent sound knowledge and are handled as the rules in LP. On the other hand, *defeasible rules* represent tentative knowledge that may be defeated by other information.

Again as in LP, DeLP operates by answering queries posed by the users. A query  $Q$  succeeds if there exists a warranted argument  $\mathcal{A}$  for  $Q$ . Arguments are constructed using both types of rules and facts (which can be seen as special cases of strict rules). The inference mechanism generates all the arguments that either support or contradict  $Q$ . Then, it runs a *warrant procedure* that determines which arguments end up undefeated. If there exists at least one argument warranting  $Q$ , it yields a positive answer.

The core of DeLP resides in the characterization of the warrant procedure. Defeasible Argumentation has provided a solid foundation over which the standard formalization of this procedure has been constructed. A key element in the warrant procedure is the criteria according to which two contradicting arguments are compared and eventually one of them deemed as defeating the other (Simari, Chesñevar, & García 1994). Pure syntactic criteria like *specificity* are both easy to understand and to implement, and therefore constituted the main choice in the design of the warrant procedure (Poole 1985; Simari & Loui 1992; Stolzenburg *et al.* 2003).

However, it seems unreasonable to limit the possibilities of comparison among arguments only to syntactic criteria. The justification of the methods of Defeasible Argumentation are largely pragmatic, that is, based on how human reasoners perform in the actual world. That is why the ultimate test for systems of defeasible reasoning is how they respond to certain benchmark problems. Therefore, it seems sensible to expand the

---

\*Partially supported by Secretaría General de Ciencia y Tecnología de la Universidad Nacional del Sur, the Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096) and the National Research Council (CONICET), Argentina

criteria of comparison to incorporate other pragmatic reasons for choosing one argument over another.

Since the issue is one of how to make *choices*, it is natural to resort to the tools of Decision Theory (DT). In fact, DT's goal is to capture in formal models the actual behavior of decision-makers. Therefore, as a discipline, it has characterized and introduced in formal models all kinds of pragmatic criteria used in actual choice situations (Loui 1998).

Any analysis in Decision Theory begins with the characterization of the *preferences* of a decision-maker, in the form of an ordering of the alternatives, one of which she has to select. Once this order is defined, the *rational* behavior of the agent is to choose one of the maximal elements. To obtain sound outcomes it is required that the ordering of alternatives be *complete* and *transitive*. This so called *weak order* can be easily represented by means of a *utility* function, that assigns to each alternative its rank (or a monotone transformation of the rank). In other terms, each alternative receives a numerical tag and the alternative with the highest value is chosen (Doyle 1990; Osborne & Rubinstein 1994).

To introduce decision-theoretic tools into DeLP we have to define what the alternatives are in this case. A hasty answer could be the arguments that can be constructed, but this is not an appropriate option. The idea of introducing this tools in DeLP is to preserve its declarative nature but expand the possible set of warrants for queries. Since the generation of arguments is the task of the inference engine, to assign utilities to them involves to reintroduce the user in a process that should remain opaque for her (Loui 1990; Tohmé 2002). Therefore, the preferences must be already defined in the characterization of a Defeasible Logic Program. That leaves only two possibilities: utilities must be attached either to rules or to facts. There are no major reasons to prefer one possibility over the other, since preferences are to be defined by the user, who may find reasons to rank both the rules and the facts to be used in the construction of arguments. Notice that if we allow utilities to be attached to rules, the distinction between strict and defeasible fades away, since every rule becomes defeasible just because it can be outranked by another rule. Again, this is a decision to be left to the user.

The plan of the rest of this paper is as follows. In section 2 we will present the rudiments of DeLP without utilities. In section 3 we introduce utilities and describe how arguments may become ranked by the inference engine. Section 4 discusses possible extensions for this work.

## Defeasible Logic Programming

In order to discuss the introduction of utilities in DeLP, we have to present the basics of this formalism (see (García & Simari 2004) for a full presentation). It has a language with three disjoint components:

- *Facts*, which are ground literals representing atomic information (or the negation of atomic information).
- *Strict Rules* of the form  $L_0 \leftarrow L_1, \dots, L_n$ , where  $L_0$  is the *head* and  $\{L_i\}_{i>0}$  is the *body*. Each  $L_i$  in the body or the head is a literal.
- *Defeasible Rules* of the form  $L_0 \multimap L_1, \dots, L_n$ , where  $L_0$  is the *head* and  $\{L_i\}_{i>0}$  is the *body*. Each  $L_i$  in the body or the head is a literal.

Then, a *Defeasible Logic Program*  $\mathcal{P}$  is a set of facts, strict rules, and defeasible rules.  $\mathcal{P} = (\Pi, \Delta)$ , where  $\Pi$  denotes the set of facts and strict rules, while  $\Delta$  denotes the set of defeasible rules. For each query  $Q$  there are four possible answers: *YES*, *NO*, *UNDECIDED* or *UNKNOWN*.

To determine which answer is correct, we need the notion of *argument*. Given a program  $\mathcal{P} = (\Pi, \Delta)$  and a literal  $L$ ,  $\langle \mathcal{A}, L \rangle$  is an argument structure for  $L$ .  $\mathcal{A}$  is a set of defeasible rules in  $\Delta$  such that:

1. there exists a *defeasible derivation* of  $L$  from  $\Pi \cup \mathcal{A}$ . That is, there exists a finite sequence  $L_1, \dots, L_n = L$  of ground literals, such that each  $L_i$  is either a fact in  $\Pi$  or there exists a rule in  $\Pi \cup \mathcal{A}$  with  $L_i$  as its head, and every literal  $B_j$  in the body is such that  $B_j \in \{L_k\}_{k<i}$ ,
2. there is no literal  $P$  such that both  $P$  and  $\neg P$  have defeasible derivations from  $\Pi \cup \mathcal{A}$ ,
3.  $\mathcal{A}$  is minimal, *i.e.*, there does not exist  $\mathcal{A}_1 \subseteq \mathcal{A}$  such that  $\mathcal{A}_1$  satisfies (1) and (2).

An argumentation process proceeds through comparisons among arguments. The main criterion of comparison used in DeLP is *specificity*. Consider a defeasible logic program  $\mathcal{P} = (\Pi, \Delta)$  with  $\Pi_G$  the set of strict rules from  $\Pi$ . Let  $\mathcal{F}$  the set of all literals that can have a defeasible derivation from  $\Pi \cup \Delta$ . Let  $\langle \mathcal{A}_1, L_1 \rangle$  and  $\langle \mathcal{A}_2, L_2 \rangle$  be two argument structures with  $L_1, L_2 \in \mathcal{F}$ . Then  $\langle \mathcal{A}_1, L_1 \rangle$  is *strictly more specific than*  $\langle \mathcal{A}_2, L_2 \rangle$  if:

- For all  $H \subseteq \mathcal{F}$ , if there exists a defeasible derivation of  $L_1$  from  $\Pi_G \cup H \cup \mathcal{A}_1$  while  $\Pi_G \cup H \not\vdash L_1$ , then  $L_2$  can be defeasibly derived from  $\Pi_G \cup H \cup \mathcal{A}_2$ , and
- There exists  $H' \subseteq \mathcal{F}$  such that there exists a defeasible derivation of  $L_2$  from  $\Pi_G \cup H' \cup \mathcal{A}_2$  and  $\Pi_G \cup H' \not\vdash L_2$  but there is no defeasible derivation of  $L_1$  from  $\Pi_G \cup H' \cup \mathcal{A}_1$ .

Argument  $\langle \mathcal{A}_1, L_1 \rangle$  *counterargues* another  $\langle \mathcal{A}_2, L_2 \rangle$  at a literal  $L$  if there exists a sub-argument of  $\langle \mathcal{A}_2, L_2 \rangle$ ,  $\langle \mathcal{A}, L \rangle$ , *i.e.*,  $\mathcal{A} \subseteq \mathcal{A}_2$ , such that there exists a literal  $P$  verifying both  $\Pi \cup \{L, L_1\} \vdash P$  and  $\Pi \cup \{L, L_1\} \vdash \neg P$ .

If  $\langle \mathcal{A}_1, L_1 \rangle$  and  $\langle \mathcal{A}_2, L_2 \rangle$  are two argument structures,  $\langle \mathcal{A}_1, L_1 \rangle$  is a *proper defeater* for  $\langle \mathcal{A}_2, L_2 \rangle$  at literal  $L$  iff there exists a sub-argument of  $\langle \mathcal{A}_2, L_2 \rangle$ ,  $\langle \mathcal{A}, L \rangle$  such that  $\langle \mathcal{A}_1, L_1 \rangle$  counterargues  $\langle \mathcal{A}_2, L_2 \rangle$  at  $L$  and  $\langle \mathcal{A}_1, L_1 \rangle$  is strictly more specific than  $\langle \mathcal{A}, L \rangle$ . Alternatively,  $\langle \mathcal{A}_1, L_1 \rangle$  is a *blocking defeater* for  $\langle \mathcal{A}_2, L_2 \rangle$  at literal  $L$  iff there exists a sub-argument of  $\langle \mathcal{A}_2, L_2 \rangle$ ,  $\langle \mathcal{A}, L \rangle$  such that  $\langle \mathcal{A}_1, L_1 \rangle$  counterargues  $\langle \mathcal{A}_2, L_2 \rangle$  at  $L$  and neither  $\langle \mathcal{A}_1, L_1 \rangle$  is strictly more specific than  $\langle \mathcal{A}, L \rangle$  nor is

$\langle \mathcal{A}, L \rangle$  strictly more specific than  $\langle \mathcal{A}, L \rangle$ . If  $\langle \mathcal{A}_1, L_1 \rangle$  is either a proper or a blocking defeater of  $\langle \mathcal{A}_2, L_2 \rangle$ , it is said to be a *defeater* of the latter.

An *argumentation line* for an argument structure  $\langle \mathcal{A}_0, L_0 \rangle$  is a sequence  $\Lambda = [\langle \mathcal{A}_0, L_0 \rangle, \langle \mathcal{A}_1, L_1 \rangle, \langle \mathcal{A}_2, L_2 \rangle, \dots]$  where for each  $i > 0$   $\langle \mathcal{A}_{i+1}, L_{i+1} \rangle$  is a defeater of  $\langle \mathcal{A}_i, L_i \rangle$ .  $\Lambda_S = [\langle \mathcal{A}_0, L_0 \rangle, \langle \mathcal{A}_2, L_2 \rangle, \langle \mathcal{A}_4, L_4 \rangle, \dots]$  is the sequence of *supporting* argument structures of  $\Lambda$ , while the sequence of *interfering* ones is  $\Lambda_I = [\langle \mathcal{A}_1, L_1 \rangle, \langle \mathcal{A}_3, L_3 \rangle, \langle \mathcal{A}_5, L_5 \rangle, \dots]$ .

An *acceptable* argumentation line in a defeasible program  $\mathcal{P} = (\Pi, \Delta)$  is a finite sequence  $\Lambda = [\langle \mathcal{A}_0, L_0 \rangle, \dots, \langle \mathcal{A}_n, L_n \rangle]$  such that:

1. Both  $\Lambda_S$  and  $\Lambda_I$  are *concordant*, i.e., there is no  $P$  such that both  $P$  and  $\neg P$  have defeasible derivations from  $\Pi \cup \bigcup_{i=0}^{\lfloor \frac{n}{2} \rfloor} \mathcal{A}_{2i}$  and no  $P'$  with defeasible derivations for both  $P'$  and  $\neg P'$  from  $\Pi \cup \bigcup_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \mathcal{A}_{2i+1}$ .
2. No argument  $\langle \mathcal{A}_k, L_k \rangle \in \Lambda$  is a subargument of an argument  $\langle \mathcal{A}_j, L_j \rangle$ , i.e.,  $\mathcal{A}_k \not\subseteq \mathcal{A}_j$ , for  $j < k$ .
3. For each  $i < n$ , if  $\langle \mathcal{A}_i, L_i \rangle$  is a blocking defeater of  $\langle \mathcal{A}_{i-1}, L_{i-1} \rangle$  then  $\langle \mathcal{A}_{i+1}, L_{i+1} \rangle$  is a proper defeater of  $\langle \mathcal{A}_i, L_i \rangle$ .

To answer a query  $Q$ , the *warrant procedure* builds up a candidate argument structure  $\langle \mathcal{A}, Q \rangle$ . Then, it associates to this argument a *dialectical tree*  $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$  as follows:

1. The root of the tree is labeled,  $\langle \mathcal{A}_0, Q_0 \rangle$ , i.e.,  $\mathcal{A}_0 = \mathcal{A}$  and  $Q_0 = Q$ .
2. Let  $n$  be a non-root node, with label  $\langle \mathcal{A}_n, Q_n \rangle$  and  $\Lambda = [\langle \mathcal{A}_0, Q_0 \rangle, \dots, \langle \mathcal{A}_n, Q_n \rangle]$  the labels in the path from the root to  $n$ . Let  $\mathbf{B} = \{\langle \mathcal{B}_1, H_1 \rangle, \dots, \langle \mathcal{B}_k, H_k \rangle\}$  be the set of all the defeaters for  $\langle \mathcal{A}_n, Q_n \rangle$ . For  $1 \leq i \leq k$ , if  $\Lambda' = [\langle \mathcal{A}_0, Q_0 \rangle, \dots, \langle \mathcal{A}_n, Q_n \rangle, \langle \mathcal{B}_i, H_i \rangle]$  is an acceptable argumentation line,  $n$  has a child  $n_i$  labeled  $\langle \mathcal{B}_i, H_i \rangle$ . If  $\mathbf{B} = \emptyset$  or no  $\langle \mathcal{B}_i, H_i \rangle \in \mathbf{B}$  is such that  $\Lambda'$  is acceptable, then  $n$  is a leaf of the tree.

The nodes of  $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$  can be marked, yielding a tagged tree  $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}^*$  as follows:

- All leaves of  $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$  are marked  $U$  in  $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}^*$ .
- If  $\langle \mathcal{B}, H \rangle$  is the label of a node which is not a leaf, the node will be marked  $U$  in  $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}^*$  if every child is marked  $D$ . Otherwise, if at least one of its children is marked  $U$ , it is marked as  $D$ .

Then, given an argument  $\langle \mathcal{A}, Q \rangle$  and its associated tagged tree  $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}^*$ , if the root is marked  $U$ , the literal  $Q$  is said to be *warranted*.  $\mathcal{A}$  is said to be the *warrant* for  $Q$ . Therefore, given a query  $Q$  the possible answers will be:

- **YES**, if  $Q$  is warranted.
- **NO**, if  $\neg Q$  is warranted.
- **UNDECIDED**, if neither  $Q$  nor  $\neg Q$  are warranted.
- **UNKNOWN**, if  $Q$  is not in the language of the program.

## Decision-Theoretic Defeat

As the quick overview of DeLP shows, the key for the warrant procedure is the characterization of the *defeat* relation among argument structures. As we have said, specificity is introduced in the standard characterization of DeLP as an example of comparison criterion among arguments. We claim that an alternative comparison criterion may arise from decision-theoretic considerations.

We will extend DeLP to allow utility values both for facts and rules. In this sense, we speak of decision-theoretic defeasible logic programs as  $\mathcal{P}' = (\Pi, \Delta, \Phi, \mathbf{B})$  where  $\Pi$  and  $\Delta$  are as before, while  $\Phi : \Pi \cup \Delta \rightarrow \mathbf{B}$ , where  $\mathbf{B}$  is an arbitrary Boolean algebra with top  $\top$  and bottom  $\perp$ . The new elements  $\Phi(\cdot)$  and  $\mathbf{B}$  represent the explicit preferences of the user, in the sense that given two pieces of information  $\mu_1, \mu_2 \in \Pi \cup \Delta$  if  $\mu_1$  is strictly *more preferred* than  $\mu_2$  then  $\Phi(\mu_1) \succ_{\mathbf{B}} \Phi(\mu_2)$ , where  $\succ_{\mathbf{B}}$  is the order of  $\mathbf{B}$ . The elements  $\mu$  of  $\Pi \cup \Delta$  which are most preferred receive a label  $\Phi(\mu) = \top$ .

We do not assume here that  $\Phi$  assigns  $\top$  to all strict rules in  $\Pi$ , and not even that  $\Phi(\mu_1) \succ_{\mathbf{B}} \Phi(\mu_2)$  for  $\mu_1 \in \Pi$  and  $\mu_2 \in \Delta$ . This is because  $\Phi(\cdot)$  has, unlike the distinction between *strict* and *defeasible* rules, no epistemic content. Instead, the preferences represent other kinds of rationales, like the reliability of the source of information that provided the rule or fact, or the cost-benefit rates of the pieces of information (since their use may preclude the use of other pieces in the reasoning process), or just the inclination towards the use of certain information over another. Examples of these attitudes are pervasive even in scientific reasoning, and we will not go further into this. Of course, nothing prevents a user from giving the highest preference to strict rules and facts.

Whatever the reasons are for preferring elements of  $\Pi \cup \Delta$ , the user has also to define the Boolean algebra  $\mathbf{B}$  over which  $\Phi(\cdot)$  ranges. It can be argued that a more general ordering could be appropriate but, as we will see, the inference engine has to perform some operations over the labels of the pieces of information used in the process of argumentation. Therefore, the range of  $\Phi(\cdot)$  has to be not only an ordered set but also an algebra. In the simplest case, in which  $\mathbf{B}$  is just a compact subset of real numbers with the natural order, we may say that  $\Phi(\mu)$  is the *utility* of the piece of information  $\mu$ .

From the preferences over  $\Pi \cup \Delta$ , we can find *preferential values* over defeasible derivations. Given a defeasible derivation of  $L$  from  $\Pi \cup \Delta$ ,  $L_1, \dots, L_n = L$ , let  $\mathcal{D}$  be the set  $\{L_1, \dots, L_n\}$  and  $\{\mu_1, \dots, \mu_n\}$  a set of rules such that  $\mu_i$  is the rule that yields  $L_i$ . Then, that derivation yields for its conclusion  $L$  a value  $V(L, \mathcal{D}) = \bigwedge_{i=1}^n V(L_i, \mu_i)$ , where  $V(L, \mu)$  is defined inductively as  $V(L, \mu) = \Phi(\mu)$  if  $L$  is a fact, i.e. the body of the rule  $\mu$  is empty, and  $V(L, \mu) = \Phi(\mu) \wedge \bigwedge_{k=1}^m V(B_k, \mu_k)$  if  $\mu$  is a rule (strict or defeasible) with head  $L$  and body  $B_1, \dots, B_m$  which is used to derive  $L$  and  $\mu_k$  is a rule used to derive  $B_k$ . The intuition here is that a conclu-

sion is as strongly preferred as the weakest of either its premises or the rule used in the derivation.

By extension, an argument structure  $\langle \mathcal{A}, L \rangle$  yields a value for  $L$ ,  $V(L, \mathcal{A}) = \bigwedge_{\mathcal{D}} V(L, \mathcal{D})$ , where  $\mathcal{D}$  is a derivation that uses all the defeasible rules in  $\mathcal{A}$  and only those defeasible rules. That is, it yields the lowest value among all the derivations of  $L$  by using defeasible rules in  $\mathcal{A}$ . Notice that, by definition of  $\mathcal{A}$  there is no other set  $\mathcal{A}' \subset \mathcal{A}$  that allows the derivation of  $L$ , but more than one selection of strict rules may exist in  $\Pi$  that allows, jointly with  $\mathcal{A}$ , to do that.

Let  $\mathcal{F}$  be the set of all literals that can have a defeasible derivation from  $\Pi \cup \Delta$ . Any subset  $H \subseteq \mathcal{F}$  has a value  $V(H) = \bigvee_{L \in H} \bigwedge_{\mathcal{D}} V(L, \mathcal{D})$ . This means that  $H$  is as valuable as the most valuable of its elements, which in turn is as valuable as the weakest of its derivations. We can use this notion to redefine *specificity* to yield a relation of *preferential specificity*. Consider again  $\Pi_G$ , the set of strict rules from  $\Pi$ . Let  $\langle \mathcal{A}_1, L_1 \rangle$  and  $\langle \mathcal{A}_2, L_2 \rangle$  be two argument structures with  $L_1, L_2 \in \mathcal{F}$ . Then  $\langle \mathcal{A}_1, L_1 \rangle$  is *strictly more preferentially specific than*  $\langle \mathcal{A}_2, L_2 \rangle$  if:

1. For all  $H \subseteq \mathcal{F}$ , if there exists a defeasible derivation of  $L_1$  from  $\Pi_G \cup H \cup \mathcal{A}_1$  while  $\Pi_G \cup H \not\vdash L_1$ , then  $L_2$  can be defeasibly derived from  $\Pi_G \cup H \cup \mathcal{A}_2$ , and
2. there exists  $H' \subseteq \mathcal{F}$  such that there exists a defeasible derivation of  $h_2$  from  $\Pi_G \cup H' \cup \mathcal{A}_2$  and  $\Pi_G \cup H' \not\vdash L_2$  but there is no defeasible derivation of  $L_1$  from  $\Pi_G \cup H' \cup \mathcal{A}_1$ .
3. For every  $H$  verifying (1) and  $H'$  verifying (2),  $V(H) \succeq_{\mathbf{B}} V(H')$ .

**Example 1** Consider a classical example in defeasible argumentation where preferences are defined for  $\mathbf{B} = \{0, 1\}$ , with  $0 < 1$ , (the preferences are indicated in parenthesis next to the corresponding pieces of information):

$$\begin{aligned} \Pi &= \{ \text{bird}(X) \prec \text{penguin}(X) \ (1), \\ &\quad \text{penguin}(\text{tweety}) \ (0), \text{bird}(\text{tweety}) \ (1) \} \\ \Delta &= \{ \neg \text{flies}(X) \prec \text{penguin}(X) \ (1), \\ &\quad \text{flies}(X) \prec \text{bird}(X) \ (1) \} \end{aligned}$$

Notice that  $\text{bird}(\text{tweety})$  yields two values:  $V(\text{bird}(\text{tweety}), \{\text{penguin}(\text{tweety}), \text{bird}(\text{tweety})\}) = \min(0, 1) = 0$  and  $V(\text{bird}(\text{tweety}), \emptyset) = 1$ , because the fact that *tweety* is a penguin has a preference of 0 while the rule used to derive that it is a bird has a preference of 1.

Consider two possible arguments:

$$\begin{aligned} &\langle \{ \neg \text{flies}(X) \prec \text{penguin}(X) \}, \neg \text{flies}(\text{tweety}) \rangle \text{ and} \\ &\langle \{ \text{flies}(X) \prec \text{bird}(X) \}, \text{flies}(\text{tweety}) \rangle \end{aligned}$$

Then, if we consider  $H = \{\text{penguin}(\text{tweety})\}$  and  $H' = \{\text{bird}(\text{tweety})\}$  we have that

$$H \cup \{ \text{bird}(X) \prec \text{penguin}(X) \} \not\vdash \neg \text{flies}(\text{tweety}),$$

but

$$\begin{aligned} &H \cup \{ \text{bird}(X) \prec \text{penguin}(X) \} \cup \\ &\quad \{ \neg \text{flies}(X) \prec \text{penguin}(X) \} \end{aligned}$$

allows the defeasible derivation of  $\neg \text{flies}(\text{tweety})$ . Furthermore,

$$H \cup \{ \text{bird}(X) \prec \text{penguin}(X) \} \cup \{ \text{flies}(X) \prec \text{bird}(X) \}$$

allows the defeasible derivation of  $\text{flies}(\text{tweety})$ .

On the other hand,

$$H' \cup \{ \text{bird}(X) \prec \text{penguin}(X) \} \not\vdash \text{flies}(\text{tweety})$$

but

$$\begin{aligned} &H' \cup \{ \text{bird}(X) \prec \text{penguin}(X) \} \cup \\ &\quad \{ \text{flies}(X) \prec \text{bird}(X) \} \end{aligned}$$

allows the defeasible derivation of  $\text{flies}(\text{tweety})$  while

$$\begin{aligned} &H \cup \{ \text{bird}(X) \prec \text{penguin}(X) \} \cup \\ &\quad \{ \neg \text{flies}(X) \prec \text{penguin}(X) \} \end{aligned}$$

does not allow the defeasible derivation of  $\neg \text{flies}(\text{tweety})$ . This implies that

$$\langle \{ \neg \text{flies}(X) \prec \text{penguin}(X) \}, \neg \text{flies}(\text{tweety}) \rangle$$

is strictly more specific than the argument  $\langle \{ \text{flies}(X) \prec \text{bird}(X) \}, \text{flies}(\text{tweety}) \rangle$ . But it is not strictly preferentially more specific, since  $V(H') = \max(V(\text{bird}(\text{tweety}), \emptyset), V(\text{bird}(\text{tweety}), \{\text{penguin}(\text{tweety}), \text{bird}(\text{tweety})\})) = \max(1, 0) = 1$  while  $V(H) = V(\text{penguin}(\text{tweety}), \emptyset) = 0$ .

A basic property of this extended version of specificity is the following:

**Proposition 1** If  $\langle \mathcal{A}_1, L_1 \rangle$  is strictly more preferentially specific than  $\langle \mathcal{A}_2, L_2 \rangle$  then  $\langle \mathcal{A}_1, L_1 \rangle$  is strictly more specific than  $\langle \mathcal{A}_2, L_2 \rangle$ .

**Proof:** Trivial: if  $\langle \mathcal{A}_1, L_1 \rangle$  is strictly more preferentially specific than  $\langle \mathcal{A}_2, L_2 \rangle$  then, it has to verify the conditions (1), (2) and (3) of the characterization of strictly-more-preferentially-specific-than relation. Since conditions (1) and (2) characterize the strictly-more-specific-than relation, the claim follows.  $\square$

A particular case in which both relations coincide is when there exists  $\mathbf{e} \in \mathbf{B}$  such that for every  $\mu \in \Pi \cup \Delta$ ,  $\Phi(\mu) = \mathbf{e}$ . But in general, the converse of Proposition 1 requires an additional property of the sets  $H, H' \subseteq \mathcal{F}$  called the *activation sets* of  $\langle \mathcal{A}_1, L_1 \rangle$  and  $\langle \mathcal{A}_2, L_2 \rangle$ , respectively. That is, that  $V(H) \succeq_{\mathbf{B}} V(H')$ . This means that:

**Proposition 2** The relation strictly-more-preferentially-specific-than in program  $\mathcal{P}' = (\Pi, \Delta, \Phi, \mathbf{B})$  is equivalent (i.e., yields the same subset of  $\text{ARG} \times \text{ARG}$  where  $\text{ARG}$  is the class of argument structures) to the relation strictly-more-specific-than in program  $\mathcal{P} = (\Pi, \Delta)$  if and only if for every pair of argument structures  $\langle \mathcal{A}_1, L_1 \rangle, \langle \mathcal{A}_2, L_2 \rangle \in \text{ARG}$ ,  $\langle \mathcal{A}_1, L_1 \rangle$  is strictly-more-specific-than  $\langle \mathcal{A}_2, L_2 \rangle$  and for every pair of their corresponding activation sets  $H, H' \subseteq \mathcal{F}$ ,  $V(H) \succeq_{\mathbf{B}} V(H')$ .

**Proof:** ( $\Rightarrow$ ): Assume that for every pair of argument structures  $\langle \mathcal{A}_1, L_1 \rangle$  is strictly-more-preferentially-specific-than  $\langle \mathcal{A}_2, L_2 \rangle$  is equivalent to  $\langle \mathcal{A}_1, L_1 \rangle$  being strictly-more-specific-than  $\langle \mathcal{A}_2, L_2 \rangle$ . Therefore, since condition (3) in the characterization of strictly-more-preferentially-specific-than must be fulfilled, for every pair of activation sets  $H$  and  $H'$  we have that  $V(H) \succeq_{\mathbf{B}} V(H')$ .

( $\Leftarrow$ ) Trivial: by definition, if  $\langle \mathcal{A}_1, L_1 \rangle$  is strictly-more-specific-than  $\langle \mathcal{A}_2, L_2 \rangle$  and for every pair of their corresponding activation sets  $H, H' \subseteq \mathcal{F}$ ,  $V(H) \succeq_{\mathbf{B}} V(H')$  we have that  $\langle \mathcal{A}_1, L_1 \rangle$  is strictly-more-preferentially-specific-than  $\langle \mathcal{A}_2, L_2 \rangle$ .  $\square$

Based on the relation strictly-more-preferentially-specific-than, we can find the derived relation of *preferential defeat*, that is obtained by replacing specificity with preferential specificity in its characterization. It follows that the warrant procedure remains the same with defeat replaced by preferential defeat. We end up having a notion of *preferential warrant* that is obtained through this procedure.

An important property of the warrant procedure is the following:

**Proposition 3** Given a query  $Q$  in the preferential defeasible logic program  $\mathcal{P}' = (\Pi, \Delta, \Phi, \mathbf{B})$ , and an argument structure  $\langle \mathcal{A}, Q \rangle$ , its tagged dialectical tree is identical to  $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}^*$  in  $\mathcal{P} = (\Pi, \Delta)$  iff the relation strictly-more-preferentially-specific-than for program  $\mathcal{P}'$  is equivalent to the relation strictly-more-specific-than in program  $\mathcal{P}$  over  $\text{ARG}^Q$ , where  $\text{ARG}^Q$  is the class of all arguments that are either labels of the dialectical tree  $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$  or subarguments of them.

**Proof:** ( $\Leftarrow$ ) Trivial.

( $\Rightarrow$ ) Suppose that  $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}^*$  is the same for both  $\mathcal{P}'$  and  $\mathcal{P}$ , but there are at least two argument structures  $\langle \mathcal{A}_1, L_1 \rangle, \langle \mathcal{A}_2, L_2 \rangle \in \text{ARG}^Q$  such that  $\langle \mathcal{A}_1, L_1 \rangle$  is strictly-more-specific-than a subargument  $\langle \mathcal{A}, L \rangle$  of  $\langle \mathcal{A}_2, L_2 \rangle$  but not strictly more preferentially specific. That means that  $\langle \mathcal{A}_1, L_1 \rangle$  is a proper defeater of  $\langle \mathcal{A}_2, L_2 \rangle$  at  $h$  but it is not a preferentially proper defeater. Therefore, the dialectical tree is different in  $\mathcal{P}'$  than in  $\mathcal{P}$ . Contradiction.  $\square$

**Corollary 1** Given a query  $Q$  and an argument structure  $\langle \mathcal{A}, Q \rangle$ , the answer to  $Q$  in the preferential defeasible logic program  $\mathcal{P}' = (\Pi, \Delta, \Phi, \mathbf{B})$  is identical to its answer in  $\mathcal{P} = (\Pi, \Delta)$  iff the relation strictly-more-preferentially-specific-than for  $\mathcal{P}'$  is equivalent to the relation strictly-more-specific-than in  $\mathcal{P}$  over  $\text{ARG}^Q$ .

## Conclusions

We presented in this paper a framework of preferential defeasible logic programming, which extends DeLP simply by redefining the relation of *specificity*, by allowing different preferential values of activation sets. Even

if an argument structure is syntactically more specific than another, it is not deemed preferentially specific until the corresponding relation of preferences holds over their activation sets.

It is a matter of further study to see how sensitive the results of DeLP are to the inclusion of preferential (utility) values. Our intuition is that quite different outcomes may be expected, although they always reflect some ordering of the information provided by the user. In either case, the final outcomes of preferential defeasible logic programming result from blending the ordering given by the user with a syntactical procedure, proper of DeLP. Also, the termination result implies the existence of a wide set of alternatives and a taxonomy of them could be part of future work.

## References

- Chesñevar, C. I.; Dix, J.; Stolzenburg, F.; and Simari, G. R. 2003. Relating defeasible and normal logic programming through transformation properties. *Theoretical Computer Science* 290(1):499–529.
- Chesñevar, C. I.; Maguitman, A. G.; and Loui, R. P. 2000. Logical Models of Argument. *ACM Computing Surveys* 32(4).
- Doyle, J. 1990. Rationality and its roles in reasoning. In *AAAI*, 1093–1100.
- 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. J. 2000. *Defeasible Logic Programming: Definition, Operational Semantics and Parallelism*. Ph.D. Dissertation, Computer Science and Engineering Department, Universidad Nacional del Sur, Bahía Blanca, Argentina.
- Loui, R. P. 1990. Defeasible specification of utilities. In Kyburg, H.; Loui, R.; and Carlson, G., eds., *Knowledge Representation and Defeasible Reasoning*. Dordrecht: Kluwer Academic Publishers. 345–359.
- Loui, R. P. 1998. Process and policy: Resource-bounded nondemonstrative reasoning. *COMPINT: Computational Intelligence: An International Journal* 14.
- Nute, D. 1994. Defeasible logic. In Gabbay, D.; Hogger, C.; and J.A. Robinson., eds., *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol 3*. Oxford University Press. 355–395.
- Osborne, M. J., and Rubinstein, A. 1994. *A Course in Game Theory*. Cambridge (MA.): MIT Press.
- Pollock, J. 1987. Defeasible Reasoning. *Cognitive Science* 11:481–518.
- Pollock, J. 1995. *Cognitive Carpentry: A Blueprint for How to Build a Person*. MIT Press.
- Poole, D. L. 1985. On the Comparison of Theories: Preferring the Most Specific Explanation. In *Proc. 9th IJCAI*, 144–147. IJCAI.

- Simari, G. R., and Loui, R. P. 1992. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence* 53:125–157.
- Simari, G. R.; Chesñevar, C. I.; and García, A. J. 1994. The role of dialectics in defeasible argumentation. In *XIV International Conference of the Chilean Computer Science Society*.
- Stolzenburg, F.; García, A. J.; Chesñevar, C. I.; and Simari, G. R. 2003. Computing generalized specificity. *Journal of Applied Non-Classical Logics* 13(1):87–113.
- Tohmé, F. 2002. Negotiation and defeasible decision making. *Theory and Decision* 53(4):289–311.