# Backing and Undercutting in Defeasible Logic Programming*

Andrea Cohen, 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 (DCIC)
Universidad Nacional del Sur (UNS), Bahía Blanca, Argentina
{ac,ajg,grs}@cs.uns.edu.ar

**Abstract.** Two important notions within the field of classical argumentation are undercutting defeaters and backings. The former represent an attack to an inference step, and the latter intend to provide defense against this type of attack. Defeasible Logic Programming (DELP) is a concrete argumentation system that allows to identify arguments whose conclusions or intermediate conclusions are in contradiction, capturing the notion of rebutting defeater. Nevertheless, in DELP is not possible to represent neither undercutting defeaters nor backings. The aim of this work is to extend the formalism of DELP to allow attack and support for defeasible rules. Thus, it will be possible to build arguments for representing undercutting defeaters and backings.

## 1 Introduction

Argumentation is a form of reasoning where a claim is accepted or rejected according to the analysis of the arguments for and against it. The way in which arguments and justifications for a claim are considered allows for an automatic reasoning mechanism where contradictory, incomplete and uncertain information may appear. In the last decade, argumentation has evolved as an attractive paradigm for conceptualizing commonsense reasoning [11]. As a consequence, several abstract argumentation frameworks and Rule-Based Argumentation Systems (RBAS) were formalized (*e. g.* [3,4,1,10,6]). Notwithstanding, a usual critique to some RBAS is that certain reasoning patterns studied in areas like legal reasoning and philosophy, which constitute important contributions to the argumentation community, were simplified or not considered in the systems formal definition. For instance, Pollock [9] stated that reasoning operates in terms of reasons that can be assembled to comprise arguments. He also established that *defeasible reasons* have defeaters, and that there are two kinds of defeaters: *rebutting defeaters* and *undercutting defeaters*. The former attack the conclusion of an inference by supporting the opposite one, while the latter attack the connection between the premises and the conclusion without attacking the conclusion directly. Another important contribution to the argumentation field was proposed by Toulmin [12]. He argued that arguments had to be analyzed using a richer format than the traditional one of formal logic. Whereas a formal logic analysis uses the dichotomy of premises and conclusions, Toulmin proposed

* Partially supported by UNS (PGI 24/ZN18) and CONICET (PIP 112-200801-02798).

W. Liu (Ed.): ECSQARU 2011, LNAI 6717, pp. 50–61, 2011.

a model for the layout of arguments that in addition to *data* and *claim* distinguishes four elements: *warrant*, *backing*, *rebuttal* and *qualifier*.

In this work, we aim to incorporate some of these elements into a concrete RBAS called Defeasible Logic Programming (DELP) [4]. Briefly, DELP is a formalism that combines argumentation and logic programming. It allows to identify arguments whose conclusions or intermediate conclusions are in contradiction, capturing Pollock's rebutting defeaters. However, as we will show, in DELP is not possible to represent neither Pollock's undercutting defeaters nor Toulmin's backings. Our proposal is partly based on [2], where a preliminary version of Extended Defeasible Logic Programing (E-DELP) was presented. The contribution of this work is to extend the formalism of DELP to capture undercutting defeaters and backings, allowing to build arguments that provide reasons for or against defeasible rules.

The rest of this paper is organized as follows. In Section 2 we present an overview of DELP and we introduce the motivation of this work. In Section 3 the extended representational language of E-DELP is proposed, and in Section 4 the notions of defeasible derivation, argument and defeater are introduced. Finally, in Section 5 some conclusions and related work are commented.

## 2 Background and Motivation

A short explanation of DELP is included below (see [4] for full details). As in Logic Programming, knowledge in DELP is represented using facts and rules. In addition, DELP has the declarative capability of representing weak information in the form of defeasible rules, and a defeasible argumentation inference mechanism for warranting the entailed conclusions.

A *defeasible logic program* (*de.l.p.*) is a set of facts, strict rules and defeasible rules, defined as follows. *Facts* are ground literals representing atomic information, or the negation of atomic information using *strong negation* "$\sim$" (*e. g.* $\sim electricity$ or $day$). *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.* $\sim night \leftarrow day$). *Defeasible Rules* represent tentative information that may be used if nothing could be posed against it 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.* $light\_on \prec switch\_on$). A defeasible rule *"$Head \prec Body$"* expresses that *"reasons to believe in the antecedent Body give reasons to believe in the consequent Head"*. When required, a defeasible logic program $\mathcal{P}$ is denoted as ($\Pi$, $\Delta$) distinguishing the subset $\Pi$ of facts and strict rules, and the subset $\Delta$ of defeasible rules. From a program $\mathcal{P}$ contradictory literals could be derived, since strong negation is allowed in the head of rules.

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. The dialectical process involves the construction and evaluation of arguments that either support or interfere with the query under analysis. Briefly, an *argument* for a literal $h$, denoted $\langle \mathcal{A}, h \rangle$, is a minimal set $\mathcal{A}$ of defeasible rules such that $\mathcal{A} \cup \Pi$ is non-contradictory and there is a derivation for $h$ from $\mathcal{A} \cup \Pi$.