Chapter 8
Argument-based Logic Programming

Alejandro J. García, Jürgen Dix and Guillermo R. Simari

1 Introduction

In this chapter we describe several formalisms for integrating Logic Programming and Argumentation. Research on the relation between logic programming and argumentation has been and still is fruitful in both directions: Some argumentation formalisms were used to define semantics for logic programming and also logic programming was used for providing an underlying representational language for non-abstract argumentation formalisms.

One of the first attempts dates back to 1987, when Donald Nute [19] introduced a formalism called LDR (Logic for Defeasible Reasoning) with a simple representational language consisting of three types of rules: Strict, defeasible and defeaters. Although LDR is not a defeasible argumentation formalism in itself, its implementation, d-Prolog, defined as an extension of PROLOG, was the first language that introduced defeasible reasoning programming with specificity as a comparison criterion between rules.

Another important step was taken in the nineties, when Phan M. Dung [10] emphasized that “there are extremely interesting relations between argumentation and logic programming”. Dung showed that argumentation can be viewed as a special form of logic programming with negation as failure. He introduced a general logic

Alejandro J. García
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Department of Computer Science and Engineering, Universidad Nacional del Sur
e-mail: ajg@cs.uns.edu.ar

Guillermo R. Simari
Department of Computer Science and Engineering, Universidad Nacional del Sur
e-mail: grs@cs.uns.edu.ar

Jürgen Dix
Department of Informatics Clausthal University of Technology
e-mail: dix@tu-clausthal.de
A few years later, in 1997, inspired by legal reasoning, Prakken and Sartor [24] introduced an argument-based formalism for extended logic programming with defeasible priorities. In their formalism, arguments are expressed in a logic programming language with both strong and default negation. Conflicts between arguments are decided with the help of priorities on the rules. These priorities can be defeasibly derived as conclusions within the system. The semantics of the system is given by a fixed point definition, while its proof theory is stated in dialectical style. A proof takes the form of a dialogue between a proponent and an opponent: An argument is shown to be justified if the proponent can make the opponent run out of moves in whatever way the opponent attacks.

Another formalism for combining logic programming and argumentation, Defeasible Logic Programming (DeLP) has been introduced by García and Simari in [15]. The representational language of DeLP is defined as an extension of a logic programming language that considers two types of rules: Strict and defeasible, and allows for both strong and default negation. A DeLP-query succeeds, i.e., it is warranted from a DeLP-program, if it is possible to build an argument that supports the query and this argument is found to be undefeated by a warrant procedure. This process implements an exhaustive dialectical analysis that involves the construction and evaluation of arguments that either support or interfere with the given query. The DeLP dialectical analysis is based on previous works on defeasible argumentation conducted by Simari and Loui [28], and Simari, Chesñevar, and García [27].

In [26], Schweimeier and Schroeder formulate a variety of notions of attack for extended logic programs from combinations of undercut and rebuttal. As shown in Section 4, their language corresponds to that used by Prakken and Sartor [24] without strict rules and any priorities. They also define a general hierarchy of argumentation semantics parameterized by the notions of attack chosen by the proponent and the opponent. They prove the equivalence and subset relationships between the semantics and examine some essential properties concerning consistency and the coherence principle, which relates default negation and explicit negation.

The rest of the chapter is organized as follows. Section 2 describes Defeasible Logic Programming. Section 3 introduces Prakken’s approach of argument-based extended logic programming with defeasible priorities. Section 4 gives an overview of argumentation semantics for extended logic programming by Schweimeier and Schroeder. Section 5 introduces Dung’s approach and Section 6 illustrates Nute’s framework. Finally, in Section 7 we discuss more recent developments before concluding with Section 8.

## 2 Defeasible Logic Programming

*Defeasible Logic Programming* (DeLP), as introduced in [15], is a formalism that combines techniques of both logic programming and defeasible argumentation. As