On the relationship between DeLP and ASPIC$^+$

Simon Parsons[1] and Andrea Cohen[2]

[1] Department of Informatics, King's College London
simon.parsons@kcl.ac.uk
[2] Institute for Computer Science and Engineering, CONICET-UNS
Department of Computer Science and Engineering, Universidad Nacional del Sur
ac@cs.uns.edu.ar

Abstract. In this chapter we consider the relationship between DeLP and ASPIC$^+$. The fact that these systems are different is well known, but what is less well known is exactly how these systems differ, and, perhaps more interestingly, the ways in which they are the similar. We do not get to the bottom of the relationship between the systems in this chapter, but we do at least set the foundations for a detailed exploration.

Keywords: structured argumentation, defeasible logic programming, DeLP, ASPIC$^+$

1 Introduction

As discussed in [8], work on argumentation can be traced back at least as far as the mid-1980s, where it grew out of attempts to create logics that were capable of defeasible reasoning. Early work on argumentation produced many different systems with different ways of representing knowledge, with different ways of constructing arguments, with different ways of identifying conflicts between arguments, and with different methods for identifying what conclusions were *acceptable*, that is which conclusions could reasonably be drawn from a given knowledge base. In the early 1990s, Dung's introduction of abstract argumentation [12, 13] changed the study of argumentation in two ways. First, Dung's work led researchers to separate the process of determining acceptable conclusions from the process of determining what arguments could be constructed from a knowledge base, and what conflicts existed between them. The first of these processes was the domain of abstract argumentation, the second of these processes was the domain of *structured argumentation*. Second, Dung's work let to a form of standardization of work on argumentation. Because his approach to determining acceptable arguments — and hence the *justified* conclusions of the acceptable arguments — became widely adopted, it led to most work on argumentation having a common theme in its use of the Dung semantics.

One indication of the extent to which the Dung semantics came to dominate work on argumentation is their adoption by the ASPIC$^+$ framework. ASPIC$^+$ [17, 20] was conceived as a general abstract model of structured argumentation, and reading through the detail of the way it represents knowledge, constructs arguments, and identifies conflicts (see Section 2), it clearly draws in elements of many different argumentation systems. However, it only considers one method for establishing which arguments are acceptable, the Dung semantics.

Now, widely used as they are, the Dung semantics are not the only way to establish the conclusions of an argumentation system. Indeed, there are approaches to doing just this which have been around longer. One of these was proposed by Guillermo Simari in his PhD thesis in 1989 [21, 22], and this method for establishing the *warranted* conclusions of a knowledge base was later integrated into the Defeasible Logic Programming (DeLP) approach proposed in [14]. Because DeLP does not make use of the Dung semantics, it cannot be thought of as a specialization of the ASPIC$^+$ framework. However, that does not, on its own, mean that the frameworks are particularly different. Indeed, as we show in this chapter, the two approaches are similar in many regards.

In the body of this chapter we seek to revisit several aspects that differentiate DeLP from ASPIC$^+$, analyze the common grounds between the two approaches, and study the possibility of establishing conditions, either on ASPIC$^+$ or on DeLP, that would help bridge the gap between them. We start with a brief introduction of the two, ASPIC$^+$ in Section 2 and DeLP in Section 3. Then, in Section 4 we move on to comparing the two approaches. We discuss the similarities and differences between them under four headings: their knowledge representation capabilities, the mechanism they adopt for argument construction, the different kinds of attack and defeat they consider, and the way in which they select accepted arguments and justified conclusions.

2 ASPIC$^+$ Background

ASPIC$^+$ is deliberately defined in a rather abstract way, as a system with a minimal set of features that can capture the notion of argumentation. This is done with the intention that it can be instantiated by a number of concrete systems that then inherit all of the properties of the more abstract system. ASPIC$^+$ starts from a logical language $\mathcal{L}$ with a notion of negation. A given instantiation will then be equipped with inference rules, and ASPIC$^+$ distinguishes two kinds of inference rules: strict rules and defeasible rules. Strict rules, denoted using $\rightarrow$, are rules whose conclusions hold without exception. Defeasible rules, denoted $\Rightarrow$, are rules whose conclusions hold unless there is an exception.

The language and the set of rules define an *argumentation system*:

Definition 1 (Argumentation System). *An* argumentation system *is a tuple* $AS = \langle \mathcal{L}, ^-, \texttt{Rules}, n \rangle$ *where:*

- $\mathcal{L}$ *is a logical language.*
- $^-$ *is a function from* $\mathcal{L}$ *to* $2^{\mathcal{L}}$ *, such that:*
  - φ *is a* contrary *of* ψ *if* $\varphi \in \overline{\psi}$*,* $\psi \notin \overline{\varphi}$*;*
  - φ *is a* contradictory *of* ψ *if* $\varphi \in \overline{\psi}$*,* $\psi \in \overline{\varphi}$*;*
  - *each* $\varphi \in \mathcal{L}$ *has at least one contradictory.*

- $\texttt{Rules} = \texttt{Rules}_s \cup \texttt{Rules}_d$ *is a set of strict* ($\texttt{Rules}_s$) *and defeasible* ($\texttt{Rules}_d$) *inference rules of the form* $\phi_1, \ldots, \phi_n \rightarrow \phi$ *and* $\phi_1, \ldots, \phi_n \Rightarrow \phi$ *respectively (where* ϕ_i, ϕ *are meta-variables ranging over wff in* $\mathcal{L}$*), and* $\texttt{Rules}_s \cap \texttt{Rules}_d = \emptyset$.
- $n : \texttt{Rules}_d \mapsto \mathcal{L}$ *is a naming convention for defeasible rules.*

The function $\bar{\ }$ generalizes the usual symmetric notion of negation to allow non-symmetric conflict between elements of $\mathcal{L}$. The contradictory of some $\varphi \in \mathcal{L}$ is close to the usual notion of negation, and we denote that φ is a *contradictory* of ψ by "$\varphi = \neg\psi$". Note that, given the characterization of $\bar{\ }$, elements in $\mathcal{L}$ may have multiple contraries and contradictories. As we will see below, the naming convention for defeasible rules is necessary because there are cases in which we want to write rules that deny the applicability of certain defeasible rules. Naming the rules, and having those names be in $\mathcal{L}$ makes it possible to do this, and the denying applicability makes use of the contraries of the rule names.

An argumentation system, as defined above, is just a language and some rules which can be applied to formulae in that language. To provide a framework in which reasoning can happen, we need to add information that is known, or believed, to be true. In ASPIC$^+$, this information makes up a *knowledge base*:

Definition 2 (Knowledge Base). *A* knowledge base *in an argumentation system* $\langle \mathcal{L}, \bar{\ }, \texttt{Rules}, n \rangle$ *is a set* $\mathcal{K} \subseteq \mathcal{L}$ *consisting of two disjoint subsets* $\mathcal{K}_n$ *and* $\mathcal{K}_p$.

We call $\mathcal{K}_n$ the axioms and $\mathcal{K}_p$ the ordinary premises. We make this distinction between the elements of the knowledge base for the same reason that we make the distinction between strict and defeasible rules. We are distinguishing between those elements — axioms and strict rules — which are definitely true and allow truth-preserving inferences to be made, and those elements — ordinary premises and defeasible rules — which can be disputed.

Combining the notions of argumentation system and knowledge base gives us the notion of an *argumentation theory*:

Definition 3 (Argumentation Theory). *An* argumentation theory *AT is a pair* $\langle AS, \mathcal{K} \rangle$ *of an argumentation system AS and a knowledge base* $\mathcal{K}$.

We are now nearly ready to define an argument. But first we need to introduce some notions that can be defined just by understanding that an argument is made up of some subset of the knowledge base $\mathcal{K}$, along with a sequence of rules, that lead to a conclusion. Given this, $\texttt{Prem}(\cdot)$ returns all the premises, $\texttt{Conc}(\cdot)$ returns the conclusion and $\texttt{TopRule}(\cdot)$ returns the last rule in the argument. $\texttt{Sub}(\cdot)$ returns all the sub-arguments of a given argument, that is all the arguments that are contained in the given argument. In addition, given $A' \in \texttt{Sub}(A)$ such that $A' \neq A$, we will say that A' is a *proper sub-argument* of A.

Definition 4 (Argument). *An* argument *A from an argumentation theory* $AT = \langle \langle \mathcal{L}, \bar{\ }, \texttt{Rules}, n \rangle, \mathcal{K} \rangle$ *is:*

1. ϕ *if* $\phi \in \mathcal{K}$ *with:* $\texttt{Prem}(A) = \{\phi\}$; $\texttt{Conc}(A) = \phi$; $\texttt{Sub}(A) = \{A\}$; *and* $\texttt{TopRule}(A) =$ *undefined.*

2. $A_1, \ldots, A_n \rightarrow \phi$ *if* A_i, $1 \leq i \leq n$, *are arguments and there exists a strict rule of the form* $\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow \phi$ *in* $\texttt{Rules}_s$. $\texttt{Prem}(A) = \texttt{Prem}(A_1) \cup \ldots \cup \texttt{Prem}(A_n)$*;* $\texttt{Conc}(A) = \phi$*;* $\texttt{Sub}(A) = \texttt{Sub}(A_1) \cup \ldots \cup \texttt{Sub}(A_n) \cup \{A\}$*; and* $\texttt{TopRule}(A) = \texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow \phi$.
3. $A_1, \ldots, A_n \Rightarrow \phi$ *if* A_i, $1 \leq i \leq n$, *are arguments and there exists a defeasible rule of the form* $\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \Rightarrow \phi$ *in* $\texttt{Rules}_d$. $\texttt{Prem}(A) = \texttt{Prem}(A_1) \cup \ldots \cup \texttt{Prem}(A_n)$*;* $\texttt{Conc}(A) = \phi$*;* $\texttt{Sub}(A) = \texttt{Sub}(A_1) \cup \ldots \cup \texttt{Sub}(A_n) \cup \{A\}$*; and* $\texttt{TopRule}(A) = \texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \Rightarrow \phi$.

We write $\mathcal{A}(AT)$ *to denote the set of arguments from the theory* AT.

In other words, an argument is either an element of $\mathcal{K}$, or it is a rule and its conclusion such that each premise of the rule is the conclusion of an argument. Note that, as stated by the authors in [17]: "Note that all premises in ASPIC$^+$ arguments are used in deriving its conclusion, so enforcing a notion of relevance analogous to the subset minimality condition requirement on premises in classical logic approaches to argumentation".

A key concept in argumentation is the idea that even if there is an argument for some conclusion, indicating that there is a *prima facie* case for the conclusion, the conclusion may not be reasonable because there is a stronger argument that it does not hold. This notion is particularly natural in a multiagent setting, where different agents have different viewpoints, leading to conflicting arguments. However, it is perfectly possible for a single argumentation theory, representing the information held by a single individual, to be the basis of conflicting arguments. We capture this kind of interaction through the idea that one argument can attack and defeat another.

An argument can be attacked in three ways: on its ordinary premises, on its conclusion (either final or intermediate), or on its defeasible inference rules. These three kinds of attack are called *undermining*, *rebutting* and *undercutting* attacks, respectively.

Definition 5 (Attack). *An argument* A attacks *an argument* B *iff* A *undermines, rebuts or undercuts* B*, where:*

– A undermines B *(on* B'*) iff* $\texttt{Conc}(A) \in \overline{\phi}$ *for some* $B' = \phi \in \texttt{Prem}(B)$ *and* $\phi \in \mathcal{K}_p$.
– A rebuts B *(on* B'*) iff* $\texttt{Conc}(A) \in \overline{\phi}$ *for some* $B' \in \texttt{Sub}(B)$ *of the form* $B_1'', \ldots, B_2'' \Rightarrow \phi$.
– A undercuts B *(on* B'*) iff* $\texttt{Conc}(A) \in \overline{n(r)}$ *for some* $B' \in \texttt{Sub}(B)$ *such that* $\texttt{TopRule}(B')$ *is a defeasible rule* r *of the form* $\phi_1, \ldots, \phi_n \Rightarrow \phi$.

We denote "A attacks B" by (A, B).

In all these cases, the idea is that an attack can be made on an element of an argument that is not known for sure to hold. An attack can thus be made on an ordinary premise — which might be an assumption or a belief — rather than an axiom, and both the other forms of attack involve defeasible rules. The difference between strict rules, using $\rightarrow$, and defeasible rules, using $\Rightarrow$, is nicely summarized by [14]. A defeasible rule captures "tentative information that may be used if nothing (can) be posed against it". The fact that "nothing can be posed against" the use of a defeasible rule is established by a proof mechanism that looks for arguments against conclusions established using defeasible rules [14]:

> (a) defeasible rule represents a weak connection between the head and the body of the rule. The effect of a defeasible rule comes from a dialectical analysis ... which involves the consideration of arguments and counter-arguments where that rule is included.

ASPIC$^+$ allows defeasible rules to be undercut, in which case the application of the rule is attacked by an argument that states the rule does not hold[3]. Similarly, since defeasible rules are tentative, ASPIC$^+$ allows the conclusions of such rules to be rebutted. The particular notion of rebutting used in ASPIC$^+$ is said to be *restricted*, meaning that an argument with a strict `TopRule`$(\cdot)$ can rebut an argument with a defeasible `TopRule`$(\cdot)$, but not vice versa. Rebutting is thus asymmetric[4].

Typically we want to model information that is believed to different degrees, and within ASPIC$^+$ we do this using a preference order over the elements of $\mathcal{R}_d$ and $\mathcal{K}_p$. The question then is how these preferences combine into an ordering $\preceq$ over arguments:

Definition 6 (Preference Ordering). *A preference ordering $\preceq$ is a binary relation over arguments, i.e., $\preceq \subseteq \mathcal{A} \times \mathcal{A}$, where $\mathcal{A}$ is the set of all arguments from an argumentation theory. Given $A, B \in \mathcal{A}$, we say A's* preference level *is less than or equal to that of B iff $A \preceq B$.*

ASPIC$^+$ does not make any assumption about the properties of the preference ordering, but as an example of a property one might use to establish $\preceq$, consider the *weakest link* principle from [17]. This assumes two pre-orderings $\leq, \leq'$ over $\mathcal{R}_d$ and $\mathcal{K}_p$ respectively, and combines them into $A \prec B$ as follows:

- the defeasible rules in A include a rule which is weaker than (strictly less than according to $\leq$) all the defeasible rules in B, and
- the ordinary premises in A include an ordinary premise which is weaker (strictly less than according to $\leq'$) all the ordinary premises in B.

$A \prec B$ is then defined as usual as $A \preceq B$ and $B \npreceq$ A.

Given $A \prec B$, we can then use this to factor the preference over arguments into the notion of attack. Attacks can be distinguished as to whether they are preference-dependent (rebutting and undermining) or preference-independent

[3] The canonical example here comes from [19] via [17], and is the rule that normally objects that appear red, are red. However, in the situation that everything is illuminated with red light, this rule no longer holds since under red light everything, including things that are not red, will appear to be red.

[4] This asymmetry is not uncontroversial, see [4, 16] for arguments against it.

(undercutting). The former succeed only when the attacker is preferred. The latter succeed whether or not the attacker is preferred.

By combining the definition of arguments, attack relation and preference ordering, we have the following definitions:

Definition 7 (Structured Argumentation Framework). *A* Structured Argumentation Framework (SAF) *is a triple* $\langle \mathcal{A}, Att, \preceq \rangle$*, where* $\mathcal{A}$ *is the set of all arguments from an argumentation theory,* Att *is the attack relation, and* $\preceq$ *is a preference ordering on* $\mathcal{A}$*.*

Definition 8 (Defeat). *A* defeats *B iff A undercuts B, or if A rebuts/undermines B on B' and A's preference level is not less than that of B' ($A \nprec B'$).*

Then the idea of an argumentation framework follows from Definitions 7 and 8.

Definition 9 (Argumentation Framework). *An* Argumentation Framework (AF) *corresponding to a structured argumentation framework* $SAF = \langle \mathcal{A}, Att, \preceq \rangle$ *is a pair* $\langle \mathcal{A}, Defeats \rangle$ *such that* $Defeats$ *is the defeat relation on* $\mathcal{A}$ *determined by* SAF*.*

In the general case, argumentation frameworks will include a defeat relation between arguments, and a natural question is what arguments are considered reasonable given those defeats. Now, argumentation frameworks as defined in Definition 9 correspond to the abstract argumentation frameworks of [13]. As a result, all the mechanisms that are defined in [13], and in later work such as [2, 5, 6, 23, 24], for establishing the *acceptability* of a set of arguments — that is identifying various mutually coherent subsets of arguments — can be employed.

Consider this example of an ASPIC$^+$ argumentation framework, adapted from [17]:

Example 1. Consider the argumentation system $AS_1 = \langle \mathcal{L}_1, \bar{\ }, \texttt{Rules}_1, n \rangle$, where:

$$\mathcal{L}_1 = \{a, b, c, d, e, f, nd, \neg a, \neg b, \neg c, \neg d, \neg e, \neg f, \neg nd\}$$

$\texttt{Rules}_1 = \mathcal{R}_{s_1} \cup \mathcal{R}_{d_1}$, with $\mathcal{R}_{s_1} = \{d, f \rightarrow \neg b\}$ and $\mathcal{R}_{d_1} = \{a \Rightarrow b; \neg c \Rightarrow d; e \Rightarrow f; a \Rightarrow \neg nd\}$, and the function $n(\cdot)$ gives $n(\neg c \Rightarrow d) = nd$. We then add the knowledge base $\mathcal{K}_1$ such that $\mathcal{K}_{n_1} = \emptyset$ and $\mathcal{K}_{p_1} = \{a; \neg c; e; \neg e\}$ to get the argumentation theory $AT_1 = \langle AS_1, \mathcal{K}_1 \rangle$. From this we can construct the arguments:

$A_1 = [a]; A_2 = [A_1 \Rightarrow b]; A_3 = [A_1 \Rightarrow \neg nd];$
$B_1 = [\neg c]; B_2 = [B_1 \Rightarrow d]; B'_1 = [e]; B'_2 = [B'_1 \Rightarrow f]; B = [B_2, B'_2 \rightarrow \neg b];$
$C = [\neg e];$

Let us call this set of arguments $\mathcal{A}_1$, so that: $\mathcal{A}_1 = \{A_1, A_2, A_3, B_1, B_2, B'_1, B'_2, B, C\}$. Note that $\texttt{Prem}(B) = \{\neg c; e\}$, $\texttt{Sub}(B) = \{B_1; B_2; B'_1; B'_2; B\}$, $\texttt{Conc}(B) = \neg b$, and $\texttt{TopRule}(B) = d, f \rightarrow \neg b$. The attacks between these arguments are shown in Figure 1 (a). These make up the set $Att_1 = \{(C, B'_1), (B'_1, C), (C, B'_2), (C, B), (B, A_2), (A_3, B_2), (A_3, B)\}$. With a preference order $\preceq$ defined by : $A_2 \prec B; C \prec B; C \prec B'_1; C \prec B'_2$, we have the structured argumentation framework $\langle \mathcal{A}_1, Att_1, \preceq \rangle$.

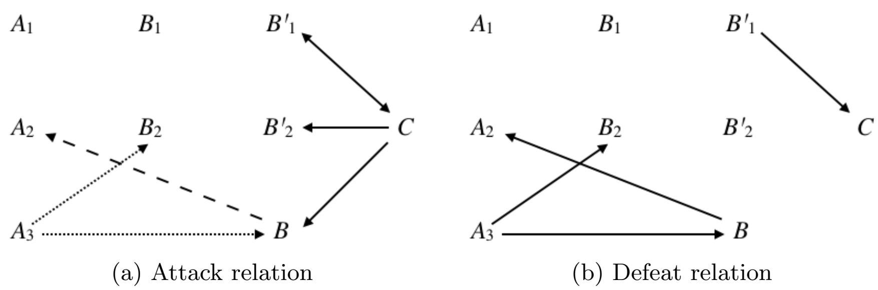


Fig. 1. Attack relations and defeat relations from Example 1. In (a), the solid arrows denote undermining attacks, the dashed arrow denotes a rebutting attack, and dotted arrows denote undercutting attacks.

This structured argumentation framework establishes a defeat relation $Defeats_1 = \{(B'_1, C), (B, A_2), (A_3, B), (A_3, B_2)\}$ which is shown in Figure 1 (b). With this, we can finally write down the argumentation framework $\langle \mathcal{A}_1, Defeats_1 \rangle$.

This completes a standard description of ASPIC$^+$. In addition, in [15] we introduced some ways of thinking about ASPIC$^+$ which will be useful here.

Definition 10. *Let $AT = \langle AS, \mathcal{K} \rangle$ be an argumentation theory, where AS is the argumentation system $AS = \langle \mathcal{L}, \bar{}, \texttt{Rules}, n \rangle$. We define the* closure *of a set of propositions $P \subseteq \mathcal{K}$ under a set of rules $R \subseteq \texttt{Rules}$ as $Cl(P)_R$, where:*

1. *$P \subseteq Cl(P)_R$;*
2. *if $p_1, \ldots, p_n \in Cl(P)_R$ and $p_1, \ldots, p_n \rightarrow p \in R_S$, then $p \in Cl(P)_R$;*
3. *if $p_1, \ldots, p_n \in Cl(P)_R$ and $p_1, \ldots, p_n \Rightarrow p \in R_D$, then $p \in Cl(P)_R$; and*
4. *$\nexists S \subset Cl(P)_R$ such that S satisfies the previous conditions.*

We use this notion of closure to establish a notion of inference in systems like ASPIC$^+$ and DeLP:

Definition 11. *Let $AT = \langle AS, \mathcal{K} \rangle$ be an argumentation theory, where AS is the argumentation system $AS = \langle \mathcal{L}, \bar{}, \texttt{Rules}, n \rangle$. Given a set of propositions $P \subseteq \mathcal{K}$, a set of rules $R \subseteq \texttt{Rules}$ and a proposition $p \in \mathcal{K}$, we say that p is inferred from P and R, noted as $P \vdash_R p$, if $p \in Cl(P)_R$.*

Finally, we made use of the idea of the set of rules in an argument:

Definition 12 (Argument Rules). *Let $AT = \langle AS, \mathcal{K} \rangle$ be an argumentation theory and $A \in \mathcal{A}(AT)$. We define the set of rules of A as follows:*

$$\texttt{Rules}(A) = \begin{cases} \emptyset & A \in \mathcal{K} \\ \\ \{\texttt{TopRule}(A)\} \cup \bigcup_{i=1}^{n} \texttt{Rules}(A_i) & A = A_1, \ldots, A_n \rightsquigarrow \texttt{Conc}(A) \end{cases}$$

This allows us to describe an argument A as a triple:

$$(G, R, p)$$

where $G = \texttt{Prem}(A)$ are the *grounds* on which A is based, $R = \texttt{Rules}(A)$ is the set of rules that are used to construct A from G, and $p = \texttt{Conc}(A)$ is the conclusion of A. Moreover, for any an argument (G, R, p), it holds that $G \vdash_R p$.

We can also identify the sets of strict and defeasible rules of an argument A as $\texttt{Rules}_s(A) = \texttt{Rules}(A) \cap \texttt{Rules}_s$ (respectively, $\texttt{Rules}_d(A) = \texttt{Rules}(A) \cap \texttt{Rules}_d$), where $\texttt{Rules} = \texttt{Rules}_s \cup \texttt{Rules}_d$ is the set of rules of the argumentation system AS in the argumentation theory AT. In addition, if $\texttt{Rules}_d(A) = \emptyset$, argument A is said to be *strict*; otherwise, if $\texttt{Rules}_d(A) \neq \emptyset$, A is a *defeasible* argument. Also, given $S \subseteq \mathcal{L}$, $S \vDash$ [5] ϕ iff there exists a strict argument A such that $\texttt{Conc}(A) = \phi$ and $\texttt{Prem}(A) \subseteq S$ (i.e. if there exists a strict argument for ϕ with all its premises taken from S).

3 DeLP Background

Defeasible Logic Programming (DeLP, for short) [14] is a formalism that combines results of Logic Programming and Defeasible Argumentation. As expressed by the authors in [14], DeLP extends logic programming with the possibility of representing information in the form of weak rules (referred to as *defeasible rules*) in a declarative manner. Then, it makes use of a defeasible argumentation inference mechanism to determine the warranted conclusions and, as a result, provide answers to queries.

The following description of DeLP is drawn from [14]. The basic unit in DeLP is a defeasible logic program:

Definition 13 (Defeasible Logic Program). *A Defeasible Logic Program $\mathcal{P}$, abbreviated* `de.l.p.`*, is a possibly infinite set of facts, strict rules and defeasible rules. In a program $\mathcal{P}$, we will distinguish the subset Π of facts and strict rules, and the subset Δ of defeasible rules. When required, we will denote $\mathcal{P}$ as (Π, Δ).*

The elements of a DeLP program are written in logic-programming style, where facts, strict rules and defeasible rules are defined in [14] as follows:

Definition 14 (Fact). *Let $\mathcal{L}$ be a set of ground atoms. A fact is a literal, i.e. a ground atom "A" or a negated ground atom "$\neg A$", where $A \in \mathcal{L}$ and "$\neg$" represents strong negation*[6]*.*

In particular, any pair of literals "A" and "$\neg A$" are said to be *complementary.*

[5] The authors in [17] use the symbol $\vdash$. We replaced it with $\vDash$ in order to avoid confusions with the notion of inference introduced in Definition 11.

[6] In [14] the authors use "$\sim$" to denote strong negation. However, in order to harmonize notation, in this chapter we will adopt the notation "$\neg$" introduced for ASPIC$^+$.

Definition 15 (Strict Rule). *A Strict Rule is an ordered pair, denoted "Head $\leftarrow$ Body", whose first member, Head, is a literal, and whose second member, Body, is a finite non-empty set of literals. A strict rule with head L_0 and body $\{L_1, \dots, L_n\}$ can also be written as: $L_0 \leftarrow L_1, \dots, L_n$ $(n > 0)$.*

It should be noted that, although the initial characterization of DeLP given in [14] requires defeasible rules to have a non-empty body, at the end of the paper the authors discuss some extensions for DeLP, among which they consider the inclusion of *presumptions* [18], which can be considered as "defeasible facts". Specifically, in [14] it is mentioned that:

> In our approach, a rule like "$a \prec$ " would express that "*there are (defeasible) reasons to believe in a.*"

Next, we present a generalized definition of defeasible rule, which accounts for presumptions:

Definition 16 (Defeasible Rule). *A Defeasible Rule is an ordered pair, denoted "Head $\prec$ Body", whose first member, Head, is a literal, and whose second member, Body, is a finite set of literals. A defeasible rule with head L_0 and body $\{L_1, \dots, L_n\}$ can also be written as: $L_0 \prec L_1, \dots, L_n$ $(n > 0)$. A defeasible rule with head L and empty body (i.e. a presumption) can also be written as: $L \prec$.*

Given a `de.l.p.`, we are interested in what can be derived from it:

Definition 17 (Defeasible Derivation). *Let $\mathcal{P} = (\Pi, \Delta)$ be a `de.l.p.` and L a ground literal. A defeasible derivation of L from $\mathcal{P}$, denoted $\mathcal{P} \mid\!\sim L$, consists of a finite sequence $L_1, L_2, \dots, L_n = L$ of ground literals, and each literal Li is in the sequence because:*

a) L_i is a fact in Π or a presumption in Δ; or
b) there exists a rule R_i in $\mathcal{P}$ (strict or defeasible) with head L_i and body $B_1, B_2, \dots, B_k$ and every literal of the body is an element L_j of the sequence appearing before L_i, $(j < i.)$

As [14] say:

> Given a `de.l.p.` $\mathcal{P}$, a derivation for a literal L from $\mathcal{P}$ is called "defeasible", because as we will show next, there may exist information in contradiction with L that will prevent the acceptance of L as a valid conclusion.

In other words, a defeasible derivation may contain strict rules, but a derivation that only contains strict rules is not considered to be a defeasible derivation.

The idea of a defeasible derivation is then used to define an *argument* in DeLP. Intuitively, an argument is a minimal set of rules used to derive a conclusion:

Definition 18 (Argument Structure). *Let L be a literal, and $\mathcal{P} = (\Pi, \Delta)$ a `de.l.p.`. We say that $\langle A, L \rangle$ is an* argument structure *for L, if A is a set of defeasible rules of Δ, such that:*

1. *there exists a defeasible derivation for L from $\Pi \cup A$;*
2. *the set $\Pi \cup A$ is non-contradictory; and*
3. *A is minimal: there is no proper subset A' of A such that A' satisfies conditions (1) and (2).*

From here on, we will sometimes refer to an argument structure simply as an argument. To complete the definition, we have to define the term "non-contradictory". [14] gives the definition:

Definition 19 (Contradictory Set of Rules). *A set of rules is contradictory if and only if, there exists a defeasible derivation for a pair of complementary literals from this set.*

and the paper takes the idea as applying to `de.l.p.`s as well (that is both sets of rules and facts can be non-contradictory). Moreover, the authors in [14] impose the requirement that the set Π of a `de.l.p.` $\mathcal{P}$ has to be non-contradictory. Specifically, this choice has to do with the meaning associated with facts and strict rules, as they represent domain information that is indisputable.

Note that the existence of a DeLP argument (with a non-contradictory and minimal set of rules) does not guarantee either its acceptance, or that its conclusion is justified. This is because the argument may be in contradiction with other arguments, which may in turn be accepted. The requirement that an argument is non-contradictory, does, however, rule out the fact that the argument is in conflict with itself. Furthermore, it rules out the possibility of building an argument that contradicts the strict knowledge of a `de.l.p.`.

Conflicts between arguments are characterized in [14] via the notion of *counter-argument*, which relies on the notion of *disagreement* between literals.

Definition 20 (Disagreement). *Let $\mathcal{P} = (\Pi, \Delta)$ be a `de.l.p.`. We say that two literals h_1 and h_2 disagree, if and only if the set $\Pi \cup \{h_1, h_2\}$ is contradictory.*

Definition 21 (Attack). *We say that $\langle A_1, h_1 \rangle$ counter-argues, rebuts, or attacks $\langle A_2, h_2 \rangle$ at literal h, if and only if there exists a sub-argument $\langle A, h \rangle$ of $\langle A_2, h_2 \rangle$ such that h_1 and h disagree.*

Note that attacks in DeLP can be aimed not only at the final conclusion of an argument, but also at its intermediate conclusions. Such intermediate conclusions correspond to the conclusions of its proper *sub-arguments* where, as defined in [14], an argument $\langle B, q \rangle$ is a sub-argument of $\langle A, h \rangle$ if $B \subseteq A$.

Given an attack from argument $\langle A_1, h_1 \rangle$ to $\langle A_2, h_2 \rangle$, these two arguments can be compared in order to determine which one prevails. Briefly, if argument $\langle A_2, h_2 \rangle$ is not better than $\langle A_1, h_1 \rangle$ with respect to a comparison criterion, noted $\langle A_1, h_1 \rangle \not\prec \langle A_2, h_2 \rangle$, $\langle A_1, h_1 \rangle$ will be called a defeater of $\langle A_2, h_2 \rangle$.

Definition 22 (Defeat). *Let $\langle A_1, h_1 \rangle$ and $\langle A_2, h_2 \rangle$ be two argument structures such that $\langle A_1, h_1 \rangle$ counter-argues $\langle A_2, h_2 \rangle$ at literal h. We say that $\langle A_1, h_1 \rangle$ is a defeater for $\langle A_2, h_2 \rangle$ if and only if either:*

a) *the attacked sub-argument $\langle A, h \rangle$ of $\langle A_2, h_2 \rangle$ is such that $\langle A, h \rangle \prec \langle A_1, h_1 \rangle$, in which case $\langle A_1, h_1 \rangle$ is a proper defeater of $\langle A_2, h_2 \rangle$; or* *b) the attacked sub-argument $\langle A, h\rangle$ of $\langle A_2, h_2\rangle$ is such that $\langle A, h\rangle \not\succ \langle A_1, h_1\rangle$ and $\langle A_1, h_1\rangle \not\succ \langle A, h\rangle$, in which case $\langle A_1, h_1\rangle$ is a blocking defeater of $\langle A_2, h_2\rangle$.*

Note that the second case in the previous definition does not only account for the case where the compared arguments are considered to be equivalent by the adopted comparison criterion, but also for the case where the attacking argument and the attacked sub-argument are incomparable (i.e. they are not related by the comparison criterion).

Example 2. Consider the `de.l.p.` $\mathcal{P}_2 = (\Pi_2, \Delta_2)$, with the same facts and rules as the argumentation system and knowledge base of the argumentation theory AT_1 from Example 1:

$$\Pi_2 = \left\{ \neg b \leftarrow d, f \right\} \qquad \Delta_2 = \begin{Bmatrix} a \prec & b \prec a \\ \neg c \prec & d \prec \neg c \\ e \prec & f \prec e \\ \neg e \prec & \neg nd \prec a \end{Bmatrix}$$

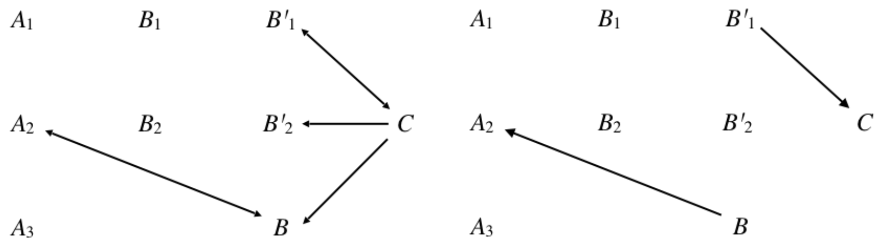
Note that the ordinary premises of the knowledge base $\mathcal{K}_1$ are represented as presumptions in Δ_2. Furthermore, in Example 1, the defeasible rule "$a \Rightarrow \neg nd$" refers to the name associated to the defeasible rule "$\neg c \Rightarrow d$" in the argumentation system AS_1. In contrast, in DeLP rules do not have associated names and thus, the literal "$\neg nd$" in the head of the defeasible rule "$\neg nd \prec a$" will not appear in any other rule or fact of the `de.l.p.` $\mathcal{P}_2$.[7]

Finally, from the DeLP program $\mathcal{P}_2$ we can build the following arguments, similarly to those obtained in Example 1:

$\langle A_1, a\rangle$, with $A_1 = \{a \prec\}$
$\langle A_2, b\rangle$, with $A_2 = \{(a \prec), (b \prec a)\}$
$\langle A_3, \neg nd\rangle$, with $A_3 = \{(a \prec), (\neg nd \prec a)\}$
$\langle B_1, \neg c\rangle$, with $B_1 = \{\neg c \prec\}$
$\langle B_2, d\rangle$, with $B_2 = \{(\neg c \prec), (d \prec \neg c)\}$
$\langle B'_1, e\rangle$, with $B'_1 = \{e \prec\}$
$\langle B'_2, f\rangle$, with $B'_2 = \{(e \prec), (f \prec e)\}$
$\langle B, \neg b\rangle$, with $B = \{(\neg c \prec), (d \prec \neg c), (e \prec), (f \prec e)\}$
$\langle C, \neg e\rangle$, with $C = \{\neg e \prec\}$

Note that the only difference between the set of ASPIC$^+$ arguments from Example 1 and the one listed above is that the DeLP argument $\langle B, \neg b\rangle$ does not include the strict rule "$\neg b \leftarrow d, f$", whereas the ASPIC$^+$ argument B includes the strict rule "$d, f \rightarrow \neg b$". Let us call the set of DeLP arguments $\mathcal{A}_2$, so that: $\mathcal{A}_2 = \{\langle A_1, a\rangle, \langle A_2, b\rangle, \langle A_3, \neg nd\rangle, \langle B_1, \neg c\rangle, \langle B_2, d\rangle, \langle B'_1, e\rangle, \langle B'_2, f\rangle, \langle B, \neg b\rangle, \langle C, \neg e\rangle\}$. The attacks between these arguments make up the relation $Att_2 = \{(\langle C, \neg e\rangle, \langle B'_1, e\rangle),$

[7] In [11] an extension of DeLP was proposed, where defeasible rules have labels associated with them, and those labels (as well as their complement with respect to the strong negation "$\neg$") can appear in the head of defeasible rules. We will come back to this point later in Section 4.3.



(a) Attack relation (b) Defeat relation

Fig. 2. (a) Attacks and (b) defeats between the arguments built from the DeLP program from Example 2.

$(\langle B_1', e\rangle, \langle C, \neg e\rangle), (\langle C, \neg e\rangle, \langle B_2', f\rangle), (\langle C, \neg e\rangle, \langle B, \neg b\rangle), (\langle B, \neg b\rangle, \langle A_2, b\rangle), (\langle A_2, b\rangle, \langle B, \neg b\rangle)\}$. This set of attacks is depicted in Figure 2 (a) where, for illustration purposes, the conclusions of the arguments are omitted. In particular, differently from Example 1, the attack involving arguments $\langle B, \neg b\rangle$ and $\langle A_2, b\rangle$ is symmetric; furthermore, since DeLP does not account for undercutting attacks, the attacks from $\langle A_3, \neg nd\rangle$ to $\langle B_2, d\rangle$ and $\langle B, \neg b\rangle$ in Example 1 no longer exist.[8] Finally, if we resolve the attacks using a comparison criterion such that: $\langle A_2, b\rangle \prec \langle B, \neg b\rangle$; $\langle C, \neg e\rangle \prec \langle B, \neg b\rangle$; $\langle C, \neg e\rangle \prec \langle B_1', e\rangle$; $\langle C, \neg e\rangle \prec \langle B_2', f\rangle$, we obtain the following defeat relation $Defeats_2 = \{(\langle B_1', e\rangle, \langle C, \neg e\rangle), (\langle B, \neg b\rangle, \langle A_2, b\rangle)\}$, shown in Figure 2 (b).

In order to determine the justified conclusions of accepted arguments built from a DeLP program (referred to in [14] as warranted literals), we need to account for the defeat relation between arguments. In particular, given an argument $\langle A_0, h_0\rangle$, all defeaters for $\langle A_0, h_0\rangle$ have to be considered. Let $\langle A_1, h_1\rangle$ be one of such defeaters. Then, since $\langle A_1, h_1\rangle$ is an argument structure, defeaters for it may also exist, and so on. As a result, in order to determine the acceptance status of argument $\langle A_0, h_0\rangle$ (thus, the warrant status of its conclusion h_0) [14] introduces the notion of *dialectical tree*, a tree structure that gathers all sequences of defeaters starting from a given argument (the root of the tree). Then, once built, the dialectical tree is *marked* according to the following criterion: (1) leaf nodes are marked as undefeated and (2) a non-leaf node (i.e. an inner node or the root) is marked as undefeated if all its children are marked as defeated; otherwise it is marked as defeated. Finally, a literal h is said to be warranted from a `de.l.p.` $\mathcal{P}$ if there exists an argument $\langle A, h\rangle$ obtained from $\mathcal{P}$ such that $\langle A, h\rangle$ is the root of a marked dialectical tree and is marked as undefeated; moreover, in that case, argument $\langle A, h\rangle$ will be considered as accepted. For full details on the

[8] In particular, the extension of DeLP introduced in [9] and [11] accounts for the existence of undercutting attacks. As mentioned before, we will come back to this point later in Section 4.3.

construction of dialectical trees and their marking criterion, we refer the reader to [14].

4 Comparison and Discussion

In this section we will study and compare different features of ASPIC$^+$ and DeLP, identifying the commonalities and differences between them. In addition, for those elements where the two systems differ, we will try to provide alternative characterizations with the aim of either bridging the gap between them, or pointing towards a way in which the gap might be bridged.

4.1 Knowledge Representation

There are clearly some commonalities between ASPIC$^+$ and DeLP. Both start with the same raw materials, a set of facts, $\mathcal{F}$, and a set of rules, $\mathcal{R}$. We can imagine both of these being partitioned into strict and defeasible parts:

$$\mathcal{F} = F_S \cup F_D$$
$$\mathcal{R} = R_S \cup R_D$$

where the subscript S denotes the strict part and D denotes the defeasible part (and there are no elements which are both strict and defeasible). Then, given sets $\mathcal{F}$ and $\mathcal{R}$, the corresponding ASPIC$^+$ argumentation system AS is $\langle \mathcal{L}, \bar{\ }, R_S \cup R_D, n \rangle$ and the corresponding knowledge base is $\mathcal{K} = F_S \cup F_D$, to then make up the argumentation theory AT $\langle AS, \mathcal{K} \rangle$. Similarly, in DeLP, this knowledge would be represented as a defeasible logic program $\mathcal{P} = (F_S \cup R_S, F_D \cup R_D)$.

Recall that, as noted above, even though the same elements are used in ASPIC$^+$ and DeLP, these systems represent defeasible facts differently. On the one hand, ASPIC$^+$ explicitly accounts for defeasible facts within the knowledge base $\mathcal{K}$: the set F_D corresponds to the set of ordinary premises $\mathcal{K}_p$. On the other hand, as mentioned before, DeLP accounts for defeasible facts under the notion of presumption; hence, defeasible facts are represented as a special case of defeasible rules. This is also how defeasible facts are represented in aspic- [4].

4.2 Argument Construction

There are several differences between the notion of argument in ASPIC$^+$ and in DeLP, and these lead to three main points of comparison:

1. Argument structure: arguments in DeLP and ASPIC$^+$ differ in structure even when built from the same knowledge, though they can be related at a rather abstract level;
2. Minimality: ASPIC$^+$ arguments are not explicitly minimal sets of facts and rules (though they contain no irrelevant information), whereas DeLP arguments are, and even when the obvious notion of minimality is applied to ASPIC$^+$ arguments it differs from that imposed in DeLP. Finally;

3. Consistency: DeLP arguments are required to conform to a kind of consistency, whereas ASPIC$^+$ arguments are not. However, ASPIC$^+$ does make use of a notion of c-consistency that can be considered to be complementary to the use of consistency in DeLP.

Argument Structure There are several ways in which the structure of DeLP and ASPIC$^+$ arguments are different.

Firstly, the two systems rely on different mechanisms for the construction of arguments. As expressed in Definition 4, an ASPIC$^+$ argument corresponds to a tree structure, where the root node is the argument itself and every other node corresponds to one of its proper sub-arguments. Furthermore, any node in the tree is connected to its children through the application of a strict or defeasible rule — specifically, the conclusions of the children are the premises of the rule and the conclusion of the parent is the conclusion of the rule. In contrast, a DeLP argument is characterized by a set of rules and a conclusion, so there is no explicit structure to a DeLP argument in the same way that there is to an ASPIC$^+$ argument. Moreover, unlike an ASPIC$^+$ argument, a DeLP argument does not contain every piece of information (facts and rules) used in its construction process. Specifically, an argument only includes the defeasible knowledge (defeasible rules and presumptions) used for building it. This does not mean that DeLP ignores strict information when constructing an argument. Rather strict knowledge is taken into account when considering the notion of defeasible derivation, a key element in the argument construction process in DeLP.

Secondly, ASPIC$^+$ does not impose restrictions on arguments, other than the implicit requirement that their construction as tree structures results from the application of strict and defeasible rules on their sub-arguments. In other words, ASPIC$^+$ arguments will not include irrelevant elements. In contrast, DeLP explicitly constrains the definition of an argument by imposing two requirements on its set of defeasible rules: minimality and consistency.

This leads us to the third difference between the ways that arguments are constructed in ASPIC$^+$ and DeLP. This is that the check that an argument is non-contradictory, performed when constructing it, effectively considers *all* the strict knowledge in the `de.l.p.`. When considering whether $\langle A, h \rangle$ is an argument, it is necessary to check that there is no combination of strict information that could, with the rules in A, be used to derive two complementary literals. This contrasts with ASPIC$^+$ which only takes into account the elements explicitly recorded in the argument structure.

Given these structural differences between arguments in ASPIC$^+$ and DeLP, we need to find a unifying mechanism for bringing them closer. For that purpose, we can consider a notion of derivation, similar to the one proposed in Definition 17 for DeLP. In particular, given an ASPIC$^+$ argument A, there exists a defeasible derivation of $\texttt{Conc}(A)$ from S, where $S = \texttt{Prem}(A) \cup \texttt{Rules}(A)$. Furthermore, the conclusion of every sub-argument of A will appear in the corresponding derivation.

However, since defeasible derivations are sequences of literals, the same argument (either in ASPIC$^+$ or in DeLP) could be associated with multiple derivations which result from the permutation of elements in the sequence (while maintaining the condition *b)* from Definition 17). Moreover, the notion of defeasible derivation makes it possible to include literals in the sequence that are not needed to derive the conclusion of an argument. Thus, by including irrelevant literals, a potentially infinite number of defeasible derivations could be associated with the same argument.

The above mentioned issue suggests the need to find an alternative unifying mechanism for ASPIC$^+$ and DeLP arguments. If we consider the sets of rules and facts used for deriving the conclusions of arguments, referred to as *deriving sets*, then we have a common basis. On the one hand, the deriving set of an ASPIC$^+$ argument A will be $\texttt{Prem}(A) \cup \texttt{Rules}(A)$. On the other hand, the deriving set of a DeLP argument $\langle A', h \rangle$ will contain every defeasible rule and presumption in A', and should also include the facts and strict rules used in the derivation of h. However, it could be the case that some literal l in the derivation of h is associated with multiple strict derivations. In such a case, the same DeLP argument would have multiple deriving sets associated with it. As a result, we can conclude that given the existence of a DeLP argument $\langle A', h \rangle$, there will exist an ASPIC$^+$ argument A with $\texttt{Conc}(A) = h$ and whose deriving set coincides with one of the deriving sets of $\langle A', h \rangle$. Conversely, given an ASPIC$^+$ argument A such that $\texttt{Conc}(A) = h$, if there exists a DeLP argument $\langle A', h \rangle$ such that A' contains the DeLP counterpart of every defeasible rule in the ASPIC$^+$ argument A, then the deriving set of A will coincide with one of the deriving sets of $\langle A', h \rangle$.

Minimality As we pointed out above, one difference between ASPIC$^+$ and DeLP is that Definition 4 does not impose any minimality requirement on the grounds or the set of rules of an ASPIC$^+$ argument while Definition 18 does. Nevertheless, as already mentioned and as discussed in [17] and [15], every premise and rule of an ASPIC$^+$ argument is used for deriving its conclusion, meaning that the argument does not contain any extraneous propositions or rules. In particular, in [15] it was formally shown that, given the characterization of an ASPIC$^+$ argument A as a triple (G, R, c), every element in G is the conclusion of a sub-argument A' of A and is the premise of a rule in R, and that every rule in R is the $\texttt{TopRule}$ of a sub-argument A'' of A. This feature of ASPIC$^+$ arguments can be considered as a form of minimality, as irrelevant elements are not introduced within an argument.

For DeLP arguments, the subset-minimality requirement imposed in the third clause of Definition 18 ensures that irrelevant elements will not be included in an argument; otherwise, there would be a smaller set of defeasible rules satisfying the first two clauses of Definition 18 and the argument in question would not be an argument. In addition, the subset-minimality requirement in DeLP avoids introducing redundant elements to an argument. To illustrate the notion of *redundancy*, let us consider the following example.

Example 3. Given the `de.l.p.` $\mathcal{P}_3 = (\Pi_3, \Delta_3)$:

$$\Pi_3 = \left\{ \begin{array}{l} p \\ q \end{array} \right\} \qquad \Delta_3 = \left\{ \begin{array}{l} r \multimap p \\ r \multimap q \\ s \multimap r \\ t \multimap r, s \end{array} \right\}$$

we can build an argument $\langle A, t \rangle$, with $A = \{(r \multimap p), (s \multimap r), (t \multimap r, s)\}$. Alternatively, we can build an argument $\langle B, t \rangle$, with $B = \{(r \multimap q), (s \multimap r), (t \multimap r, s)\}$. However, there is no argument $\langle C, t \rangle$ with $C = \{(r \multimap p), (r \multimap q), (s \multimap r), (t \multimap r, s)\}$, because C is a superset of A and B.

Let us now consider the knowledge represented in Example 3 in the context of an ASPIC$^+$ argumentation system:

Example 4. Consider the argumentation system $AS_4 = \langle \mathcal{L}_4, \bar{\ }, \mathcal{R}_4, n \rangle$, where $\mathcal{L}_4 = \{p, q, r, s, t, \neg p, \neg q, \neg r, \neg s, \neg t\}$ and $\mathcal{R}_4 = \{p \Rightarrow r; q \Rightarrow r; r \Rightarrow s; r, s \Rightarrow t\}$. By adding the knowledge base $\mathcal{K}_4 = \{p, q\}$ we obtain the argumentation theory $AT_4 = \langle AS_4, \mathcal{K}_4 \rangle$, from which we can construct (among others) the following arguments:

$$A_1 = [p]; A_2 = [A_1 \Rightarrow r]; A_3 = [A_2 \Rightarrow s]; A = [A_2, A_3 \Rightarrow t];$$
$$B_1 = [q]; B_2 = [B_1 \Rightarrow r]; B_3 = [B_2 \Rightarrow s]; B = [B_2, B_3 \Rightarrow t];$$
$$C = [B_2, A_3 \Rightarrow t]$$

Argument C from Example 4 is redundant because its set of rules provides two ways to derive r, one that relies on p and another that relies on q, and r appears twice in the derivation of t: once to produce s, and once when the rule $r, s \Rightarrow t$ is applied. Then C, the redundant argument, uses both rules for deriving r while arguments A and B use just one of them, providing a more compact derivation.

Another situation that can occur in ASPIC$^+$, which in DeLP is prevented by the third clause of Definition 18 is the existence of *circularity* within an argument. This is illustrated by the following example:

Example 5. Consider the argumentation system $AS_1 = \langle \mathcal{L}_5, \bar{\ }, \mathcal{R}_5, n \rangle$, where $\mathcal{L}_5 = \{a, b, c, d, e, \neg a, \neg b, \neg c, \neg d, \neg e\}$ and $\mathcal{R}_5 = \{a, b \Rightarrow d; d \Rightarrow b; b, c \Rightarrow e\}$. By adding the knowledge base $\mathcal{K}_5 = \{a, b, c\}$ we obtain the argumentation theory $AT_5 = \langle AS_5, \mathcal{K}_5 \rangle$, from which we can construct the following arguments:

$$D_1 = [a]; D_2 = [b]; D_3 = [D_1, D_2 \Rightarrow d]; D_4 = [D_3 \Rightarrow b]; D_5 = [c];$$
$$D = [D_4, D_5 \Rightarrow e]; E = [D_2, D_5 \Rightarrow e]$$

such that $D = (\{a, b, c\}, \{a, b \Rightarrow d; d \Rightarrow b; b, c \Rightarrow e\}, e)$ and $E = (\{b, c\}, \{b, c \Rightarrow e\}, e)$.

In the context of Example 5, argument E is circular because it starts with a and b as premises, from which it derives c. Then, it uses c to (again) derive b and, finally, use c and the second derivation of b to obtain the conclusion e. This loop is removed in E to give a more compact argument for the conclusion e.

Moreover, this circularity becomes evident when observing the characterization of D and E as a triple, since the sets of grounds and rules of argument E are proper subsets of those of argument D.

In contrast to these examples of circularity and redundancy in ASPIC$^+$, because of the subset-minimality requirement established by the third clause in Definition 18, an argument like D would not exist in DeLP.

One way to bridge this gap between ASPIC$^+$ and DeLP is to impose some form of minimality on an ASPIC$^+$ argument. In [15] we showed that defining ASPIC$^+$ arguments to eliminate circularity and redundancy was equivalent to enforcing minimality on the set of grounds or rules used to construct the argument. (To be precise, if we describe an ASPIC$^+$ argument A as a triple (G, R, c), then if there is no argument (G', R, c) such that $G' \subset G$ and no argument (G, R', c) such that $R' \subset R$, then A is not redundant or circular.) However, note that this is a less restrictive form of minimality than the one enforced in DeLP, since DeLP necessarily requires the set of defeasible rules, $R_D \subseteq R$ in the notation we introduced in Definition 10, to be minimal. Furthermore, as grounds (facts) are not included within a DeLP argument, the minimality check on that set is not necessary. As a result, it could be the case that an argument is minimal in ASPIC$^+$ but not in DeLP. In contrast, for every argument in DeLP (which, by definition, is minimal) there exists a minimal argument in ASPIC$^+$ having the same set of defeasible rules. This difference between the notion of minimality in ASPIC$^+$ and in DeLP is illustrated in the following example.

Example 6. Let $AT_6 = \langle AS_6, \mathcal{K}_6 \rangle$ be an argumentation theory, where $AS_6 = \langle \mathcal{L}_6, ^-, \mathcal{R}_6, n \rangle$, $\mathcal{R}_6 = \{d \Rightarrow b; b \Rightarrow c; b, c \Rightarrow a\}$ and $\mathcal{K}_6 = \mathcal{K}_{n_6} = \{b, d\}$. From AT we can construct the following arguments:

$$A_1 = [d]; A_2 = [A_1 \Rightarrow b]; A_3 = [A_2 \Rightarrow c]; A_4 = [b]; A = [A_4, A_3 \Rightarrow a];$$
$$B = [A_2, A_3 \Rightarrow a]; A_5 = [A_4 \Rightarrow c]; C = [A_4, A_5 \Rightarrow a]$$

Here, $A = (G, R, a)$, with $G = \{b, d\}$ and $R = \mathcal{R}_6$. In this case, A is not minimal since there exists $B = (G', R, a)$ with $G' = \{d\} \subset G$. On the other hand, argument C is represented by the triple (G'', R', a), with $G'' = \{b\}$ and $R' = \{b \Rightarrow c; b, c \Rightarrow a\}$. In particular, argument C is minimal. Furthermore, B is also minimal since, even though $R' \subset R$, (G', R', a) is not an argument for a. Let us now consider the DeLP program $\mathcal{P}_6 = (\Pi_6, \Delta_6)$:

$$\Pi_6 = \{(b \prec d), (c \prec b), (a \prec b, c)\}$$

Here, there is only one argument whose conclusion is the literal a; this is the argument $\langle C', a \rangle$, with $C' = \{(c \prec b), (a \prec b, c)\}$. In particular, argument $\langle C', a \rangle$ in DeLP would correspond to argument C in ASPIC$^+$. Note that, even though there exists a derivation for the literal a from the set $B' = \{d, (b \prec d), (c \prec b), (a \prec b, c)\}$, there is no other argument for a. In particular, B' would correspond to the ASPIC$^+$ argument B, whose set of grounds is $G' = \{d\}$ and its set of rules (following DeLP's notation) is $R = \{(b \prec d), (c \prec b), (a \prec b, c)\}$.

This relates back to the point we made in the previous section about relating DeLP and ASPIC$^+$ through deriving sets. Even if the sets of defeasible rules in a DeLP argument A and an ASPIC$^+$ argument A' coincide, the arguments can have different conclusions, or the same conclusion and different deriving sets, because the strict part of the DeLP argument is not constrained. These situations are illustrated in the following example.

Example 7. Consider the argumentation system $AS_7 = \langle \mathcal{L}_7, \bar{\ }, \mathcal{R}_7, n \rangle$, where $\mathcal{L}_7 = \{a, b, c, d, \neg a, \neg b, \neg c, \neg d\}$ and $\mathcal{R}_7 = \{a \rightarrow c; a \rightarrow b; b \rightarrow c; d \Rightarrow a\}$. Then, if we add the knowledge base $\mathcal{K}_7 = \mathcal{K}_{n_7} = \{d\}$, we get the argumentation theory $AT_7 = \langle AS_7, \mathcal{K}_7 \rangle$. From this theory, we can build the following arguments:

$$D = [d];\ \ A = [D \Rightarrow a];\ \ C = [A \rightarrow c];\ \ B = [A \rightarrow b];\ \ C' = [B \rightarrow c]$$

The DeLP counterpart of the argumentation theory AT_7 would be the `de.l.p.` $\mathcal{P}_7 = (\Pi_7, \Delta_7)$:

$$\Pi_7 = \left\{ \begin{array}{l} d \\ c \leftarrow a \\ c \leftarrow b \\ b \leftarrow a \end{array} \right\} \Delta_7 = \left\{ a \prec d \right\}$$

from which we can build the following arguments: $\langle D, d \rangle$, with $D = \emptyset$; and $\langle A, a \rangle$, $\langle A, b \rangle$, $\langle A, c \rangle$, with $A = \{a \prec d\}$. Note that, unlike in ASPIC$^+$, since DeLP arguments do not include the strict knowledge used in the derivation of their conclusions, there is only one argument for c. Furthermore, whereas in ASPIC$^+$ we have two arguments for c, one of which has a unique deriving set, argument $\langle A, c \rangle$, will have two deriving sets: one of them including the strict rule $c \leftarrow a$, and the other including the strict rules $c \leftarrow b$ and $b \leftarrow a$. As a result, the DeLP argument $\langle A, c \rangle$ has the same conclusion and the same set of defeasible rules as the ASPIC$^+$ arguments C and C', but their deriving sets differ. On the other hand, even though the DeLP argument $\langle A, a \rangle$ has the same set of defeasible rules as the ASPIC$^+$ arguments C and C', their conclusions differ: $\texttt{Conc}(C) = \texttt{Conc}(C') = c$ while the conclusion of $\langle A, a \rangle$ is a; also, the deriving set of $\langle A, a \rangle$ differs from that of the ASPIC$^+$ arguments C and C'.

Consistency The last aspect to consider in terms of argument construction is the requirement in DeLP that the set $\Pi \cup A$ which gives rise to the defeasible derivation behind an argument $\langle A, h \rangle$ is non-contradictory. The requirement for the basis of an argument to be consistent is not uncommon in argumentation systems — see, for example, [1, 3] — but this is not exactly what is required in DeLP. In DeLP, the consistency is between the argument structure in the form of the defeasible rules A, and the entire set of strict information in the knowledge base Π. The effect of the consistency requirement in DeLP, which is encoded in clause 2 of Definition 18, is to prevent an argument from coming into existence if it derives the complement of something that is in the strict part of the knowledge base Π, or can be derived from the corresponding program $\mathcal{P}$

using the facts and rules in both Π and A. That is a rather stronger check than is imposed in systems such as [1, 3], as shown in the following example:

Example 8. Let $AS_8 = \langle \mathcal{L}_8, \bar{\ }, \mathcal{R}_8, n \rangle$ be an argumentation system, where $\mathcal{L}_8 = \{a, b, \neg a, \neg b\}$ and $\mathcal{R}_8 = \{b \Rightarrow \neg a\}$. Then, if we add the knowledge base $\mathcal{K}_8 = \mathcal{K}_{n_8} = \{a, b\}$, we get the argumentation theory $AT_8 = \langle AS_8, \mathcal{K}_8 \rangle$. In ASPIC$^+$, and many other argumentation systems, from this argumentation theory we can construct the following arguments:

$$A_1 = [a];\ A_2 = [b];\ A_3 = [A_2 \Rightarrow \neg a]$$

However from the corresponding `de.l.p.` $\mathcal{P}_8 = (\Pi_8, \Delta_8)$:

$$\Pi_8 = \{a, b\} \quad \Delta_8 = \{\neg a \mathrel{-\!\!\prec} b\}$$

we can build the arguments $\langle \emptyset, a \rangle$ and $\langle \emptyset, b \rangle$, but are prevented from building the argument $\langle \{\neg a \mathrel{-\!\!\prec} b\}, \neg a \rangle$ because its conclusion $\neg a$ contradicts the fact a.

This difference represents a fundamental difference in viewpoint between DeLP and ASPIC$^+$ (and other systems like those cited above). In ASPIC$^+$, there is a clear separation between argument construction and argument evaluation. ASPIC$^+$ specifies how to construct arguments, and how to recognize conflicts (attacks and defeats) between them. This results in an argumentation framework of arguments and defeats which is then processed in exactly the same way as a Dung abstract argumentation framework [13]. Thus ASPIC$^+$ handles the above example by constructing A_1, A_2 and A_3, recognizing the conflict between A_1 and A_3, and, with its restricted rebut, only recognizing the attack of A_1 on A_3. This will result in A_1 and its conclusion being justified, whereas A_3 and its conclusion are not (under any semantics).

In contrast, the consistency checking aspect of DeLP can be seen as a combination of argument construction and evaluation. Since, like ASPIC$^+$, DeLP privileges strict information in the sense that it cannot be overturned by arguments with a defeasible component, DeLP refuses to allow arguments that (would) challenge this information to be brought into existence. In terms of the fate of A_3, this leaves ASPIC$^+$ and DeLP drawing the same conclusion — A_3 is not justified (in particular, in DeLP, because it would not even exist).

However, even though not included in Definition 4, this does not mean that ASPIC$^+$ ignores any notion of consistency *within* arguments. In [17] the authors introduce a notion of *c-consistency* that accounts only for contradictories (and not contraries); hence, it refers to "contradictory-consistency".

Definition 23 (c-consistency). *A set $S \subseteq \mathcal{L}$ is c-consistent if $\nexists \phi \in \mathcal{L}$ such that $S \vDash \phi, \neg\phi$. Otherwise S is c-inconsistent.*

Given the above definition, if $S \vDash \phi, \psi$, where $\psi \in \overline{\phi}$ but $\phi \notin \overline{\psi}$, then S can still be c-consistent. Then, the authors of [17] characterize a special class of ASPIC$^+$ arguments, whose premises are *c-consistent*.

Definition 24. *An argument A built from an argumentation theory AT on the basis of an argumentation system $\langle \mathcal{L}, \bar{\ }, \mathcal{R}, n \rangle$ and a knowledge base $\mathcal{K}$ is c-consistent iff* `Prem`(A) *is c-consistent.*

Note that a c-consistent argument is one with c-consistent premises. Thus c-consistency prevents the construction of arguments where the foundations, the premises, disagree amongst themselves by including a proposition and its contrary. In addition, because the premises are c-consistent, it is not possible to construct strict arguments from those premises such that the conclusions of those arguments contradict one another. Note that, an argument A with premise a and conclusion $\neg a$ may still be c-consistent, as long as there are no strict rules leading to derive contradictory conclusions, starting from the premise a. Indeed, c-consistency does not exclude the construction of an argument that contradicts something in the set of axioms (the strict part of the knowledge base). Thus, it is perfectly reasonable in ASPIC$^+$ to have a in the set of axioms and also have a strict argument for $\neg a$. Such an argument would not be permitted in DeLP, because the conclusion conflicts with the strict premise a. As an example of this, take Example 8 and make the defeasible rule strict. In such a case, argument $A'_3 = [A_2 \rightarrow \neg a]$ would still be c-consistent since it is not possible to obtain strict arguments with contradictory conclusions such that their premises are taken from $\{b\}$; specifically, even though the set $\{b\}$ makes it possible to obtain the strict argument A'_3, it is not possible to build A_1 (the strict argument whose conclusion contradicts $\texttt{Conc}(A'_3)$ such that its premises are taken from the set $\{b\}$. Then, because of the definition of attack in ASPIC$^+$, since the conclusion of A_1 is an axiom and $\texttt{TopRule}(A'_3)$ is strict, neither of the two arguments will attack the other, and so both a and $\neg a$ would be in the set of justified conclusions. This is exactly why such a theory is not *well-defined* [17], which is a requirement for the theory to obey the rationality postulates [7] (which require justified conclusions to be consistent).

Finally, we can conclude that the notion of c-consistency in ASPIC$^+$ is somehow complementary to the consistency check made in DeLP's argument construction process. On the one hand, DeLP avoids building an argument whose set of (defeasible) rules, together with the strict knowledge of a `de.l.p.`, leads to the derivation of complementary literals. On the other hand, in a setting where only c-consistent arguments are allowed, ASPIC$^+$ prevents the construction of an argument whose set of premises, together with other strict knowledge in the theory, leads to the construction of strict arguments with contradictory conclusions.

To summarize, both DeLP and ASPIC$^+$ perform some kind of consistency check on the arguments against the strict knowledge of the program/argumentation theory: the former by considering the defeasible rules of an argument, and the latter by considering the premises of an argument. In both cases, the check is not of the consistency of the argument itself, but of the consistency of things that can be derived from it.

4.3 Attacks and Defeats

In this section we will start by contrasting the ways in which DeLP and ASPIC$^+$ account for the existence of conflicts between arguments. Then, we will turn to analyze the mechanism in which they resolve those conflicts, leading to the existence of defeats.

As we briefly mentioned in Section 3, there are some differences in the characterization of attacks in DeLP and in ASPIC$^+$. The main difference relies on the fact that ASPIC$^+$ distinguishes between undermining, rebutting and undercutting attacks. In contrast, DeLP defines a general notion of attack, which accounts for all the possible situations in which two arguments are considered to be conflicting. This difference becomes evident when looking at the attack relation in Examples 1 and 2; moreover, this can be easily observed when contrasting Figures 1(a) and 2(a). In the following, we will discuss the three kinds of attack from ASPIC$^+$, and study ways in which they can be realized in DeLP.

Recall that undermining attacks in ASPIC$^+$ are aimed at attacking the ordinary premises of an argument. On the other hand, even though undermining attacks are not explicitly accounted as such, they are contemplated within the DeLP's general notion of attack . As discussed in Section 4.1, ordinary premises in DeLP are represented in the form of presumptions. Then, when considering Definition 21, if the attacked sub-argument $\langle A, h\rangle$ is such that its conclusion corresponds to a presumption "$h \prec$ ", then the attack would be an undermining attack. To illustrate this, let us consider again Examples 1 and 2. In particular, the undermining attacks from C to B'_1, B'_2 and B, and the undermining attack from B'_1 to C in ASPIC$^+$ still occur in DeLP.

Let us now consider rebutting attacks. To start with, even though DeLP does not explicitly distinguish between different kinds of attack, the general notion proposed in Definition 21 accounts for the situation in which rebutting attacks occur; that is, where the conclusion of the attacking argument is in conflict with the conclusion of a sub-argument of the attacked argument. However, there are several differences between the consideration of rebutting attacks in Definitions 5 and 21. On the one hand, rebutting attacks in ASPIC$^+$ are *restricted* in the sense that an argument cannot be attacked at the conclusion of a sub-argument whose `TopRule` is strict. In contrast, DeLP allows arguments to be attacked on the conclusions of strict rules, as long as the attacked sub-arguments make use of defeasible knowledge as well. In other words, once defeasibility is introduced within an argument, DeLP allows an attack at any literal whose derivation goes beyond the consideration of strict knowledge. This difference is evidenced in Examples 1 and 2, where ASPIC$^+$ only accounts for an attack from argument B to argument A_2, whereas in DeLP the arguments $\langle A_2, b\rangle$ and $\langle B, \neg b\rangle$ attack each other.

Note that restricted rebut was introduced [7] to ensure that the rationality postulate of closure (under strict rules) holds and one way to view what it does is to prioritize the conclusions obtained through the use of strict rules. (In the context of closure, if you prioritize the conclusions of strict rules, then inferences drawn from justified conclusions using strict rules will also be justified conclusions, and closure holds.) The idea of prioritization of strict rules is somehow accounted for in DeLP by imposing restrictions on the construction of arguments. Specifically, no argument in DeLP can be such that, when considered together with the strict knowledge of a `de.l.p.` allows to derive complementary literals. Then, we could say that DeLP also prioritizes the consideration of strict over defeasible knowledge. Nevertheless, we should remark that the notion of attack in DeLP can be easily modified in order to restrict rebut, similarly to ASPIC$^+$.

Another difference between the characterization of rebutting attack in ASPIC$^+$ and the general notion of attack in DeLP regards the nature of the conflict between the conclusions of the attacking argument and the attacked sub-argument. In DeLP, the existence of an attack depends on the condition that the conclusions of these two arguments disagree (i.e. when considering the two literals together with the strict knowledge of the corresponding DeLP program, complementary literals can be derived). On the other hand, ASPIC$^+$ identifies the existence of a conflict if the conclusion of the attacking argument is a contrary of the conclusion of the attacked sub-argument. Then, we can clearly identify two differences. First, ASPIC$^+$'s notion of conflict is more general, in the sense that it allows for non-symmetric attacks. That is, it can be the case that there exists an argument A that rebuts another argument B on B', but neither B nor B' rebut A, even in the case where $\texttt{TopRule}(A)$ is defeasible. In contrast, since attacks in DeLP rely on the notion of disagreement, which is inherently symmetric (i.e. if literal L_1 disagrees with literal L_2, then L_2 also disagrees with L_1), if argument $\langle A_1, L_1\rangle$ attacks argument $\langle A_2, L_2\rangle$ on the sub-argument $\langle A, h\rangle$, then it holds that $\langle A, h\rangle$ also attacks $\langle A_1, L_1\rangle$. Second, in ASPIC$^+$, the conflict between the conclusion of the attacking argument and the attacked sub-argument is always direct, as the former is a contrary of the latter. However, in DeLP, in cases where the conclusions of the attacking and the attacked sub-argument are not complementary, the conflict between them would not be direct. As a result, there might be arguments considered to be conflicting in ASPIC$^+$ (leading to the existence of a rebutting attack) which are not accounted as such in DeLP and, conversely, there might be arguments considered to be in conflict in DeLP (leading to the existence of an attack) but not in ASPIC$^+$. To illustrate this, let us consider the following example:

Example 9. Consider the argumentation system $AS_9 = \langle \mathcal{L}_9, \bar{}, \mathcal{R}_9, n\rangle$, where $\mathcal{L}_9 = \{a, b, c, d, e, f, \neg a, \neg b, \neg c, \neg d, \neg e, \neg f\}$, the contrariness function is such that $a \in \overline{f}$, and $\mathcal{R}_9 = \{a \rightarrow c; b \rightarrow \neg c; d \Rightarrow a; e \Rightarrow b; e \Rightarrow f\}$. Then, if we add the knowledge base $\mathcal{K}_9 = \mathcal{K}_{n_9} = \{d, e\}$, we get the argumentation theory $AT_9 = \langle AS_9, \mathcal{K}_9\rangle$. From this theory, we can build the following arguments:

$$D = [d]; \quad E = [e]; \quad A = [D \Rightarrow a]; \quad B = [E \Rightarrow b];$$
$$F = [E \Rightarrow f] \quad C = [A \rightarrow c]; \quad C' = [B \rightarrow \neg c]$$

Note that the only attack that arises from the consideration of these arguments is the rebutting attack from A to F (because $a \in \overline{f}$). In contrast, even though C and C' have contradictory conclusions, both arguments have a strict `TopRule`, so neither of them rebuts the other.

If we want to represent the knowledge within the argumentation theory AT_9 in DeLP, because the strong negation "$\neg$" is symmetric, we will not be able to model that a is a contrary of f. So, we have two alternatives: ignore the conflict between a and f, or represent the conflict through the use of a rule like $\neg f \prec a$, which leads the literals a and f to disagree (thus, the conflict between them to become symmetric). As the second alternative loses the intuition behind the notion of *contrary* in ASPIC$^+$, we will adopt the first one and define the `de.l.p.` $\mathcal{P}_9 = (\Pi_9, \Delta_9)$:

$$\Pi_9 = \{d, e, (c \leftarrow a), (\neg c \leftarrow b)\} \quad \Delta_9 = \{(a \prec d), (b \prec e), (f \prec e)\}$$

From this DeLP program, we can build the following arguments: $\langle D, d\rangle$ and $\langle E, e\rangle$, with $D = E = \emptyset$; $\langle A, a\rangle$ and $\langle A, c\rangle$, with $A = \{a \prec d\}$; $\langle B, b\rangle$ and $\langle B, \neg c\rangle$, with $B = \{b \prec e\}$; and $\langle F, f\rangle$, with $F = \{f \prec e\}$. Note that arguments $\langle A, c\rangle$ and $\langle B, \neg c\rangle$ would correspond to arguments C and C' in ASPIC$^+$; in particular, the sets of defeasible rules of the two DeLP arguments coincide with the sets of defeasible rules of the two corresponding ASPIC$^+$ arguments. However, unlike the two ASPIC$^+$ arguments, the two DeLP arguments will attack each other. On the other hand, since the conflict between a and f in ASPIC$^+$ is not captured within $\mathcal{P}_9$, argument $\langle A, a\rangle$ will not attack argument $\langle F, f\rangle$.

Regarding the nature of conflicts in ASPIC$^+$ and in DeLP, more specifically, the existence of contraries, it is worth to note the following. As mentioned in [17], the notion of contrary is somehow associated with the notion of *negation as failure*. Then, that a is a contrary of f (and not a contradictory) can be interpreted as $f = \mathit{not}(a)$, where "*not*" represents negation as failure. On the other hand, in [14] the authors discuss an extension of DeLP that accounts for this kind of negation (i.e. negation as failure). As a result, ASPIC$^+$'s contraries could be represented in the extended version of DeLP by making use of negation as failure. In particular, it would be possible to represent that $a \in \overline{f}$ in $\mathcal{P}_9$: the literal "f" would have to be replaced with "not(a)" and the defeasible rule "$f \prec e$" would be replaced with "$\mathit{not}(a) \prec e$". Moreover, in such a case, argument $\langle F', \mathit{not}(a)\rangle$, with $F' = \{\mathit{not}(a) \prec e\}$ would be attacked by argument $\langle A, a\rangle$.

Let us now focus on undercutting attacks. As introduced in Section 2, ASPIC$^+$ includes a naming function for defeasible rules within the characterization of an argumentation system. Then, by having the names of defeasible rules in the logical language $\mathcal{L}$, it is possible to have contraries and contradictories for them, leading to the existence of undercutting attacks. In contrast, the formalization of DeLP given in [14] does not account for the existence of undercutting attacks. Notwithstanding this, there exists an extension of DeLP that incorporates undercut as a type of attack between arguments [10, 11]. In [11] the set of atoms in a program includes a set of *labels*, and each defeasible rule has an associated label with the restriction that no pair of defeasible rules within a program shares the same label. Then, by allowing labels (and their negations with respect to "$\neg$") to appear in the head of other defeasible rules, it is possible to express reasons for and against the use of the corresponding defeasible rules. As a result, an argument whose conclusion is "$\neg l$", with "l" being the label of a defeasible rule R, would undercut every other argument including the rule R. A different approach is taken in [9]. This incorporates backing and undercutting rules as meta-rules, allowing to express reasons for and against the use of defeasible rules, respectively. Then, undercutting rules are used for building undercutting arguments, which lead to the existence of undercutting attacks.

Finally, having compared the types of attack accounted for (either explicitly or implicitly) in DeLP and ASPIC$^+$, let us turn our attention to the way in which the they determine the success of attacks, leading to the existence of defeats. Since it was shown that DeLP does not consider undercutting attacks, we will leave those out of the discussion. When contrasting Definitions 8 and 22, we see that both ASPIC$^+$ and DeLP make use of a preference ordering or a comparison criterion between arguments. Furthermore, even though DeLP distinguishes between two kinds of defeat (namely, proper and blocking defeat) whereas ASPIC$^+$ does not, they both consider the existence of a defeat if and only if the attacked sub-argument is not better than the attacking argument (under the adopted preference ordering or comparison criterion). As a result, we can conclude that DeLP and ASPIC$^+$ handle the resolution of undermining and rebutting attacks into defeats equivalently.

4.4 Acceptance of Arguments and Justification of Conclusions

This section will focus on contrasting the ways in which DeLP and ASPIC$^+$ select the accepted arguments and the justified conclusions, thus determining the inferences of the system.

As briefly introduced in sections 2 and 3, ASPIC$^+$ and DeLP adopt different approaches for this purpose. ASPIC$^+$ first constructs a Dung-like argumentation framework [13] consisting of the arguments and defeats obtained from a given argumentation theory. Then, by applying any of the existing semantics for Dung's abstract argumentation frameworks (see [2]) ASPIC$^+$ identifies the *extensions* of the framework, which correspond to collectively acceptable sets of arguments. The justified conclusions of the original theory can then be identified. In contrast, DeLP defines its own semantics based on a dialectical process that involves the construction and marking of dialectical trees. As a result, the accepted arguments built from a `de.l.p.` will be those marked as "undefeated" in their dialectical trees. Again, the conclusions can then be established from the arguments that are computed to be accepted.

One major difference that we identify between the two approaches is that ASPIC$^+$'s approach is oriented at determining the acceptance/justification status of *every* argument, and hence conclusion, in the argumentation theory. In contrast, as expressed in [14], DeLP is conceived as a query-answering tool, and its dialectical process is aimed at determining the warrant status of a queried literal "l"; hence, it only requires to consider (and analyze the acceptance status of) the arguments belonging to the dialectical tree rooted in arguments of the form $\langle A, l\rangle$ or $\langle A, \neg l\rangle$. Of course, in DeLP it is also possible to determine the acceptance status of every argument built from a `de.l.p.` and, consequently, the warrant status of every literal in that program. Specifically, the accepted arguments will be those that are marked as "undefeated" in their dialectical trees, and the warranted literals will be the accepted arguments' conclusions.

Another difference is that, given a query for a literal "l" in a program, DeLP will unequivocally provide an answer (i.e. the answer will always be the same): *YES* if "l" is warranted from the corresponding DeLP program, *NO* if the literal "$\neg l$" is warranted instead, *UNDECIDED* if neither "l" nor "$\neg l$ are warranted, or *UNKNOWN* if "l" is not in the language of the program (i.e. neither "l" nor "$\neg l$" appear in the facts or rules of the `de.l.p.`). In particular, the answer *UNDECIDED* will correspond to situations in which there are no accepted arguments for the literals "l" and "$\neg l$"; an example of this situation would be the case where there exist two arguments $\langle A, l\rangle$ and $\langle B, \neg l\rangle$ that are blocking defeaters of each other, and none of them has any other defeater. Thus, we can consider DeLP's dialectical process to be *cautious*. On the other hand, since ASPIC$^+$ allows the use of any semantics defined for Dung's AFs, it can be the case that the adopted semantics (e.g. preferred semantics) makes it possible to obtain multiple extensions and thus allows for multiple answers to a query about a given literal. For instance, in a situation like the one described above, ASPIC$^+$ would obtain the preferred extensions $\{A'\}$ and $\{B'\}$, where A' and B' would be the ASPIC$^+$ counterpart of the DeLP arguments $\langle A, l\rangle$ and $\langle B, \neg l\rangle$. Thus, one can think of the preferred semantics as offering a choice between the alternative of accepting one argument or the other and, consequently, justifying one conclusion or the other. Note that, given the existence of multiple extensions under the adopted semantics, it is possible to make use of the *credulous* or the *skeptical* (or *cautious*) acceptance of arguments: an argument will be skeptically accepted under a semantics iff it belongs to every extension obtained under that semantics, whereas it will be credulously accepted under that semantics iff it belongs to some (but not every) extension. Nevertheless, some semantics that can be used with ASPIC$^+$, like the *grounded* semantics, do not allow for multiple extensions, and thus, they can be considered to be *cautious* as well[9].

To illustrate the way in which ASPIC$^+$ and DeLP determine the accepted arguments and the justified conclusions of the system, let us consider the arguments and defeats from Examples 1 and 2.

Example 10. The arguments and defeats identified in Example 1, depicted in Figure 1(b), define the abstract argumentation framework $\langle\{A_1, A_2, A_3, B_1, B_2, B'_1, B'_2, B, C\}, \{(A_3, B), (A_3, B_2), (B, A_2), (B'_1, C)\}\rangle$. For instance, if we consider the grounded semantics, the only grounded extension will be $\{A_1, A_2, A_3, B_1, B'_1, B'_2\}$; in particular, the grounded semantics accepts all arguments having no defeaters (in this case, A_1, A_3, B_1, B'_1 and B'_2), leaves out arguments that are defeated by the undefeated arguments (here, B, B_2 and C), and includes the arguments that are defended by those already included in the extension (A_2). Therefore, the set of ASPIC$^+$ justified conclusions will be $\{a, b, \neg nd, \neg c, e, f\}$.

Let us now consider the arguments and defeats identified in Example 2 for DeLP, depicted in Figure 2(b). There, the only arguments having defeaters are

[9] In the labelling approach [2], the grounded labelling is the one that maximizes the UNDEC labels, and so, in a sense, is the most cautious of the possible semantics.

$\langle A_2, b\rangle$ (whose defeater is $\langle B, \neg b\rangle$) and $\langle C, \neg e\rangle$ (whose defeater is $\langle B', e\rangle$). Therefore, every argument except those two will be marked as "undefeated" in their dialectical trees, whereas $\langle A_2, b\rangle$ and $\langle B, \neg b\rangle$ will be marked as "defeated". As a result, the set of accepted arguments will be $\{\langle A_1, a\rangle, \langle A_3, \neg nd\rangle, \langle B_1, \neg c\rangle, \langle B_2, d\rangle, \langle B'_1, e\rangle, \langle B'_2, f\rangle, \langle B, \neg b\rangle\}$, and the set of warranted literals from the `de.l.p.` program specified in Example 2 will be $\{a, \neg d, c, d, e, f, \neg b\}$.

It can be noted that the accepted arguments and the justified/warranted conclusions in DeLP and ASPIC$^+$ differ. In particular, the difference in this case relies on the consideration of undercutting attacks. Since A_3 undercuts B_2 and B in ASPIC$^+$, argument A_2 is defended by A_3 and, as a result, can be accepted. Furthermore, arguments B and B_2 are not accepted. In contrast, since DeLP does not account for undercutting attacks, arguments $\langle A_3, \neg nd\rangle$, $\langle B_2, d\rangle$ and $\langle B, \neg b\rangle$ will be accepted together. This difference is also observed when considering the sets of justified/warranted conclusions: whereas ASPIC$^+$ justifies "b", DeLP warrants "$\neg b$".

It should be noted that, even though the difference in the results obtained by DeLP and ASPIC$^+$ in the previous example is related to the existence of undercutting attacks/defeats in ASPIC$^+$ (which do not occur in DeLP), it can also be the case that an ASPIC$^+$ theory and a `de.l.p.` have the same set of arguments and defeats but different justified/warranted conclusions. An example of such a situation would be the one described in the paragraph before Example 10, where DeLP would have no accepted arguments and no warranted conclusions (literals), whereas ASPIC$^+$ (under the preferred semantics) will consider the arguments A' and B' to be credulously accepted, and their conclusions to be credulously justified.

Finally, notwithstanding the above discussed differences in the acceptance or justification process adopted by DeLP and ASPIC$^+$, as well as the difference in their consideration of attacks and defeats, there are cases in which the two systems behave alike and their outcomes coincide. This is illustrated by the following example:

Example 11. Consider the argumentation system $AS_{11} = \langle \mathcal{L}_{11}, {}^-, \mathcal{R}_{11}, n\rangle$, where $\mathcal{L}_{11} = \{a, b, c, d, f, g, h, k, \neg a, \neg b, \neg c, \neg d, \neg f, \neg g, \neg h, \neg k\}$ and $\mathcal{R}_{11} = \{b, d \rightarrow a; h \rightarrow f; h, c \rightarrow \neg k; g \rightarrow d; \neg f \Rightarrow b; c, k \Rightarrow \neg b; g \Rightarrow k; d \Rightarrow h\}$. Then, if we add the knowledge base $\mathcal{K}_{11} = \mathcal{K}_{n_{11}} \cup \mathcal{K}_{p_{11}}$, with $\mathcal{K}_{n_{11}} = \{g, c\}$ and $\mathcal{K}_{p_{11}} = \{\neg f\}$, we get the argumentation theory $AT_{11} = \langle AS_{11}, \mathcal{K}_{11}\rangle$. From this theory, we can build the following arguments:

$F' = [\neg f]$;	$B = [F' \Rightarrow b]$;	$G = [g]$;	$D = [G \rightarrow d]$;
$A = [B, D \rightarrow a]$;	$C = [c]$;	$K = [G \Rightarrow k]$;	$B' = [C, K \Rightarrow \neg b]$;
$H = [D \Rightarrow h]$;	$K' = [H, C \rightarrow \neg k]$;	$F = [H \rightarrow f]$	

The DeLP counterpart of the argumentation theory AT_{11} would be the `de.l.p.` $\mathcal{P}_{11} = (\Pi_{11}, \Delta_{11})$:

$$\Pi_{11} = \left\{ \begin{array}{ll} g & a \leftarrow b, d \\ c & f \leftarrow h \\ & \neg k \leftarrow h, c \\ & d \leftarrow g \end{array} \right\} \Delta_{11} = \left\{ \begin{array}{l} b \prec \neg f \\ \neg b \prec c, k \\ k \prec g \\ h \prec d \\ \neg f \prec \end{array} \right\}$$

from which we can build the following arguments:

$\langle F', \neg f \rangle$, with $F' = \{\neg f \prec \}$;
$\langle B, b \rangle$, with $B = \{(b \prec \neg f), (\neg f \prec)\}$;
$\langle G, g \rangle$, with $G = \emptyset$;
$\langle D, d \rangle$, with $D = \emptyset$;
$\langle A, a \rangle$, with $A = \{(b \prec \neg f), (\neg f \prec)\}$;
$\langle C, c \rangle$, with $C = \emptyset$;
$\langle K, k \rangle$, with $K = \{k \prec g\}$;
$\langle B', \neg b \rangle$, with $B' = \{(\neg b \prec c, k), (k \prec g)\}$;
$\langle H, h \rangle$, with $H = \{h \prec d\}$;
$\langle K', \neg k \rangle$, with $K' = \{h \prec d\}$;
$\langle F, f \rangle$, with $F = \{h \prec d\}$

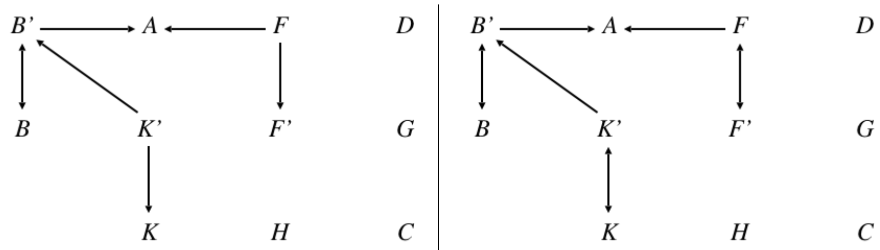
Recall that, unlike arguments in ASPIC$^+$, arguments in DeLP only include the defeasible component of the argument (i.e. defeasible rules, including presumptions). As a result, there exist different arguments, with different conclusions, that have the same associated set of rules. Nevertheless, without loss of generality, we can identify each of the argument structures listed above through the name of its associated set of rules and presumptions; hence, we can refer to them as F', B, G, $\ldots$, H, K', F, respectively.

The attacks between the arguments obtained from the argumentation theory AT_{11} are depicted in Figure 3(a), whereas the attacks between the arguments built from the `de.l.p.` $\mathcal{P}_{11}$ are illustrated in Figure 3(b). Note that the difference between the two relies on the fact that ASPIC$^+$ considers restricted rebut, and so some rebutting attacks that are symmetrical in DeLP are not symmetrical in ASPIC$^+$.

Suppose now that we consider a preference criterion or ordering on arguments such that: $B \prec B'$, $K \prec K'$ and $F' \prec F$. Then, as shown in Figure 4, the defeats between arguments built from the argumentation theory AT_{11} and the `de.l.p.` $\mathcal{P}_{11}$ coincide. Furthermore, if we consider the grounded, preferred or stable semantics (in the case of ASPIC$^+$) and DeLP's dialectical process, we obtain the same outcome: the set of accepted arguments is $\{D, G, C, H, F, K', B\}$ and the set of warranted (DeLP) and justified (ASPIC$^+$) conclusions is $\{d, g, c, h, f, \neg k, b\}$.

5 Conclusion

This chapter has examined the relationship between DeLP and ASPIC$^+$. While the analysis has been largely informal, we hope that it is still clear that the two



(a) ASPIC$^+$ attacks (b) DeLP attacks

Fig. 3. Attacks between arguments from Example 11 in (a) ASPIC$^+$ and (b) DeLP.

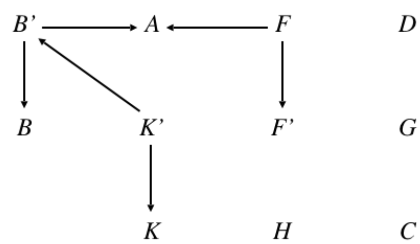


Fig. 4. Defeats between arguments built from AT_{11} and $\mathcal{P}_{11}$.

approaches are very similar in many regards. Indeed, they are perhaps more similar than they are dissimilar. In our view there is certainly enough similarity to justify a more formal analysis that looks to find out, precisely where the approaches overlap, and where they differ, especially in terms of what conclusions they draw from a given knowledge base. This is work we hope to carry out in the near future.

Acknowledgment This research was partially supported by the UK Engineering & Physical Sciences Research Council (EPSRC) under grant #EP/P010105/1. In addition, both the authors acknowledge a considerable debt to Guillermo Simari.

SP: As someone who hasn't worked directly with Guillermo, I am very grateful for the fact that he introduced me to two people who have become valued collaborators: Andrea Cohen, my co-author here, and Gerardo Simari, with whom I have worked on a number of topics over the years. I particularly value the fact that in both cases Guillermo presented the opportunity for me to work with Andrea and Gerardo as an instance of me doing him a favour, when in fact it was him who did me the favour. However, I think that my greatest debt to Guillermo is in the example that he sets. I have always found him to be among the most thoughtful and gracious people that I have met in my research career. In that

respect I think he embodies qualities that I strive (but regularly fail) to achieve myself, and I thank him for the ongoing example.

AC: I will always be grateful to Guillermo for being a wonderful teacher, supervisor and mentor. Every since I started my research career he was there for me, giving me advice and helping me (and everyone in our group) in every step of the way. Also, I am grateful to him for always pushing me to achieve great things, giving me the opportunity to live wonderful experiences such as my research stay in Brooklyn College-CUNY, where I had the pleasure to meet and start working with Simon Parsons. Last but not least, I will always cherish the thoughtful and kind words he had towards me, especially in moments when I felt I was not cut out for this. For all this, I feel lucky to have been able to grow and work with him. Thank you, Guillermo, for being our role-model.

References

1. L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artifical Intelligence*, 34(3):197–215, 2002.
2. Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4):365–410, 2011.
3. P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128:203–235, 2001.
4. M. Caminada, S. Modgil, and N. Oren. Preferences and unrestricted rebut. *Computational Models of Argument: Proceedings of COMMA 2014*, pages 209–220, 2014.
5. M. W. A. Caminada. On the issue of reinstatement in argumentation. In *Proceedings of the 10th European Conference on Logic in Artificial Intelligence*, pages 111–123, Liverpool, UK, 2006.
6. M. W. A. Caminada. An algorithm for computing semi-stable semantics. In *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 222–234, Verona, Italy, 2007.
7. M. W. A. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5–6):286–310, 2007.
8. C. I. Chesñevar, A. G. Maguitman, and R. P. Loui. Logical models of argument. *ACM Computing Surveys*, 32(4):337–383, 2000.
9. Andrea Cohen, Alejandro J. García, and Guillermo R. Simari. Backing and undercutting in defeasible logic programming. In *11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 50–61, 2011.
10. Andrea Cohen, Alejandro J. García, and Guillermo R. Simari. Backing and undercutting in abstract argumentation frameworks. In *7th International Symposium on Foundations of Information and Knowledge Systems*, pages 107–123, 2012.
11. Andrea Cohen, Alejandro Javier García, and Guillermo Ricardo Simari. Extending delp with attack and support for defeasible rules. In *IBERAMIA*, volume 6433 of *Lecture Notes in Computer Science*, pages 90–99. Springer, 2010.
12. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambéry, France, 1993.
13. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

14. A. J. García and G. Simari. Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.
15. Z. Li, A. Cohen, and S. Parsons. Two forms of minimality in ASPIC$^+$. In *15th European Conference on Multi-Agent System*, Évry, France, 2017.
16. Zimi Li and Simon Parsons. On argumentation with purely defeasible rules. In *Scalable Uncertainty Management - 9th International Conference, SUM 2015, Québec City, QC, Canada, September 16-18, 2015. Proceedings*, pages 330–343, 2015.
17. S. Modgil and H. Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.
18. Donald Nute. Defeasible reasoning: a philosophical analysis in PROLOG. In J. H. Fetzer, editor, *Aspects of Artificial Intelligence*, pages 251–288. Kluwer Academic Pub., 1988.
19. J. L. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.
20. H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.
21. G. R. Simari. *A Mathematical Treatment of Defeasible Reasoning and its Implementation*. PhD thesis, Department of Computer Science, Washington University in St Louis, 1989.
22. G. R. Simari and R. P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53:125–157, 1992.
23. B. Verheij. A labeling approach to the computation of credulous acceptance in argumentation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 623–628, Hyderabad, India, 2007.
24. G. Vreeswijk. An algorithm to compute minimally grounded and admissible defence sets in argument systems. In *Proceedings of the First International Conference on Computational Models of Argument*, pages 109–120, Liverpool, UK, 2006.