A Default Logical Semantics for Defeasible Argumentation

Gabriele Kern-Isberner
Department of Computer Science
University of Technology Dortmund, Germany

Guillermo R. Simari
Department of Computer Science & Engineering
Universidad Nacional del Sur, Argentina

Abstract

Defeasible argumentation and default reasoning are usually perceived as two similar, but distinct approaches to commonsense reasoning. In this paper, we combine these two fields by viewing ( defeasible resp. default) rules as a common crucial part in both areas. We will make use of possible worlds semantics from default reasoning to provide examples for arguments, and carry over the notion of plausibility to the argumentative framework. Moreover, we base a priority relation between arguments on the tolerance partitioning of system Z and obtain a criterion phrased in system Z terms that ensures warranty in defeasible argumentation.

Key Words: Defeasible argumentation, default reasoning, system Z, possible worlds

1 Introduction

Argumentation techniques have gained increasing relevance in artificial intelligence research during the past two decades, mainly due to their wide range of applicability within agent systems, providing bases both for single agent reasoning and deliberation, and for coordination and exchange of information in multiagent systems. Basically, argumentation investigates how arguments attack and defend one another, thereby pondering in a dialectical way the pros and cons of options to make up a proper decision. Presumably the most foundational and abstract framework of argumentation has been provided by Dung (Dung 1995), but lots of other more constructive approaches to argumentation have been brought forward as well, making use of logical elements like rules and deduction, and of priority relations to decide between good and worse arguments (see, e.g., (Besnard and Hunter 2008)).

Defeasible Logic Programming (DELP) (García and Simari 2004) combines logic programming with defeasible argumentation, allowing the representation of tentative knowledge and leaving for the argumentation mechanism the task of finding those conclusions for which it is not possible to find valid reasons against. DELP works in a highly dialectical way, allowing series of attacks and counterattacks to finally mark those statements as warranted for which all attackers could be invalidated. Although DELP rests on logic programming grounds and its warrant procedure corresponds nicely to human argumentation behaviour, it has not yet been possible to characterize its semantics of warrant along established formal logical lines (Thimm and Kern-Isberner 2008).

In this paper, we take an alternative semantical way to support the conclusions drawn by DELP. First, we introduce possible worlds as examples and counterexamples of arguments, and characterize crucial notions like attack and concordance in DELP in these terms. Then, we make use of ordinal conditional functions (Spohn 1988) that provide degrees of (dis)belief to possible worlds in order to assign priority information to arguments. The basic idea here is that arguments are as convincing and successful as their most plausible examples. So, arguments with more plausible examples should prevail, in particular when compared to counterarguments and counterexamples.

Then, in order to sharpen our results, we make use of the distinguished system Z approach (Goldszmidt and Pearl 1996) as a particularly well-behaved ordinal conditional function to assign plausibility to examples of arguments and prioritize arguments by Z-specificity values. We prove a sufficient condition to ensure warrant in DELP, and show that our combined Z-DELP-approach is able to solve the so-called drowning problem.

The rest of this paper is organized as follows: Section 2 recalls the background needed for ordinal conditional functions, system Z and DELP. Then, our semantics based on examples of arguments is developed in section 3, and is connected to plausibility in section 4. Afterwards, section 5 presents the combined Z-DELP-approach, and section 6 concludes this paper. All notions and techniques are illustrated by a running example.

2 Formal and methodological background

Logical prerequisites. Let \( \mathcal{L} \) be a finitely generated propositional language with atoms \( a, b, c, \ldots \), and with formulas \( A, B, C, \ldots \). For conciseness of notation, we will omit the logical and-connector, writing AB instead of \( A \land B \), and overlining formulas will indicate negation, i.e. \( \overline{A} \) means \( \neg A \). Let \( \Omega \) denote the set of possible worlds over \( \mathcal{L} \); \( \Omega \) will be taken here simply as the set of all propositional interpretations over \( \mathcal{L} \). For \( \omega \models A \) means that the propositional formula

Copyright © 2011, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
A ∈ Ł holds in the possible world ω ∈ Ω. The set of models of A is denoted by Mod(A) = {ω ∈ Ω | ω ⊨ A}.

By making use of a new binary operator |∼, we obtain the set (Ł | Ł) = {(B | A) | A, B ∈ Ł} of conditionals over Ł. In the framework of default reasoning, (B | A) formalizes “if A then B is plausible” and establishes a defeasible connection between the antecedent A and the consequent B. In the framework of argumentation, (B | A) may be read as “if there are reasons to believe A, then there are reasons to believe B”. Here, conditionals are supposed not to be nested, that is, antecedent and consequent of a conditional will be propositional formulas. A conditional (B | A) is an object of a three-valued nature, partitioning the set of worlds Ω in three parts: those worlds satisfying AB and thus verifying the conditional, those worlds satisfying AB, thus falsifying the conditional, and those worlds not fulfilling the premise A and so which the conditional may not be applied to at all (cf. (DeFinetti 1974)).

OCFs and system Z. Ordinal conditional functions (OCFs, also called ranking functions) are functions κ : Ω → ℕ ∪ {∞} with κ(0) = 0, expressing degrees of plausibility of propositional formulas A by specifying degrees of disbelief of their negations A (cf. (Sohn 1988; Goldszmidt and Pearl 1996)). More formally, we have κ(A) := min{κ(ω) | ω ⊨ A}, so that κ(A ∨ B) = min{κ(A), κ(B)}. A conditional (B | A) is accepted in the epistemic state represented by κ, or κ satisfies (B | A), written as κ ⊨ (B | A), iff κ(AB) < κ(A), i.e., iff AB is more plausible than A. A set Δ = {δi = (Bi | Ai) | 1 ≤ i ≤ n} of conditionals is said to be consistent iff there exists an OCF κ that accepts all conditionals in Δ, i.e., for which holds κ(AiBi) < κ(AiBi), 1 ≤ i ≤ n. Consistency can be checked easily by applying the notion of tolerance. A conditional (B | A) is tolerated by a set of conditionals Δ iff there is a world ω such that ω verifies (B | A) and ω does not falsify any of the conditionals in Δ. Δ is consistent, iff there is an ordered partition Δo, Δ1, . . . , ΔΔ of such that each conditional in Δm is tolerated by ∪j=m+1j Δj, 0 ≤ m ≤ k (Goldszmidt and Pearl 1996).

A well-known method to compute an ordinal conditional function which accepts all conditionals in a (finite) set Δ = {δi = (Bi | Ai) | 1 ≤ i ≤ n} is system Z of Goldszmidt and Pearl ((Goldszmidt and Pearl 1996)). System Z makes use of the tolerance partitioning described above, where each subset Δi of the partition is maximal, i.e., contains as many conditionals as possible. Then to each δ ∈ Δ, a rank is associated by setting Z(δ) = j iff δ ∈ Δj. The system Z ranking function, κz, accepting all conditionals in Δ is then given by

\[ κ_z(ω) = \begin{cases} 0, & \text{if } ω \text{ does not falsify any } δ_i, \\ 1 + \max_{1 ≤ i ≤ n} Z(δ_i), & \text{otherwise} \end{cases} \]  

(1) κz assigns to each world ω the lowest possible rank admissible with respect to the constraints in Δ. System Z can be used to define a nonmonotonic inference relation ∼z: Given some consistent set of conditionals Δ and propositional formulas A, B, we have A ∼z B iff κz ⊨ (B | A), i.e., iff in the context of A, B turns out to be plausible. For more details see, for instance, (Goldszmidt and Pearl 1996). We illustrate default reasoning with system Z in the following benchmark example taken from (García and Simari 2004).

Example 1 We consider the propositional variables b bird, p penguin, c chicken, s is scared, f flies, w has wings, and the set of conditionals: \[ Δ = \{δ_i = (b | c), δ_2 = (b | p), δ_3 = (f | b), δ_4 = (f | c), δ_5 = (f | s), δ_7 = (w | b)\} \]. The tolerance partitioning used by system Z is Δ0 = {δ3, δ2}, Δ1 = {δ1, δ2, δ4, δ5}, Δ2 = {δ6}. We compute κz(pbf) = 2 > 1 = κz(pbf), so penguin-birds do not fly, as expected. Also scared penguins do not fly, since κz(psφ) = 2 > 1 = κz(psf).

The basic problem of default reasoning is to decide if some formula A is a plausible consequence of a set of conditionals, Δ, given some prerequisite B. This amounts to checking if the conditional (B | A) is accepted on the ground of Δ. Ranking functions and, in particular, system Z provide a convenient semantics to verify this inference relation.

Defeasible argumentation. We will present the basics of the formalism of DELP, following the presentation in (García and Simari 2004). A defeasible logic program (de.l.p.) P = (Π, Δ) consists of a set Π of facts and strict logic rules, and a set Δ of defeasible rules which are written as conditionals (L1 | B1 . . . Bn) with literals L1, B1, . . . , Bn in this paper. Let Δ ≤ Δ be a subset of Δ. A literal L can be defeasibly derived from Δ, iff there exists a finite sequence L1, . . . , Ln = L of ground literals, such that each Li is either a fact in Π or there exists a rule in Π ∪ Δ with head Li and body {B1, . . . , Bm}, and every literal Bj in the body is such that Bj ∈ {Lk}k<k. Π ∪ Δ is called contradictory iff there is a literal L such that both L and ¬L have defeasible derivations from Π ∪ Δ. For any de.l.p. P we will presuppose that Π is non-contradictory.

Given a de.l.p. P = (Π, Δ) and a literal L, A is an argument for L, denoted (A, L), if A is a set of defeasible rules in Δ such that:

1. there exists a defeasible derivation of L from Π ∪ A;
2. Π ∪ A is non-contradictory;
3. there is no A′ ⊆ A such that A′ satisfies (1) and (2), i.e., in that sense A is minimal.

An argument ⟨B, Q⟩ is a sub-argument of ⟨A, L⟩ if B is subset of Π.

Argument ⟨A1, L1⟩ attacks, or counterargues another ⟨A2, L2⟩ at a literal L if there exists a sub-argument of ⟨A2, L2⟩, (A, L), i.e., A ⊆ A2, such that there exists a literal P verifying both Π ∪ {L, L1} ⊢ P and Π ∪ {L, L1} ⊢ P. Note that an argument ⟨∅, L⟩ with L ∈ Π can not be attacked since all arguments have to be consistent with Π.

Example 2 We will look at example 1 in the DELP-framework. Let propositional and conditionals be given as specified in example 1. Consider the defeasible logic program P1 = (⟨cs⟩, Δ). Then the following arguments can
be built supporting \( f \) resp. \( \overline{f} \):

\[
\langle A_1, f \rangle, \quad A_1 = \{ (b|c), (f|b) \}; \\
\langle A_2, \overline{f} \rangle, \quad A_2 = \{ (\overline{f}|c) \}; \\
\langle A_3, f \rangle, \quad A_3 = \{ (f|c) \}.
\]

Clearly, \( \langle A_2, \overline{f} \rangle \) attacks \( \langle A_1, f \rangle \), and \( \langle A_3, f \rangle \) attacks \( \langle A_2, \overline{f} \rangle \).

An argumentation process proceeds through comparisons among arguments. The standard criterion of comparison used in DELP is specificity, but any criterion that establishes preference could be used. Since we will use other preference criteria here, we will not go into further details here.

If \( \langle A_1, L_1 \rangle \) and \( \langle A_2, L_2 \rangle \) are two arguments \( \langle A_1, L_1 \rangle \) is a proper defeater for \( \langle A_2, L_2 \rangle \) at literal \( L \) iff there exists a sub-argument of \( \langle A_2, L_2 \rangle, \langle A, L \rangle \) such that \( \langle A_1, L_1 \rangle \) counterargues \( \langle A_2, L_2 \rangle \) at \( L \). Alternatively, \( \langle A_1, L_1 \rangle \) is a blocking defeater for \( \langle A_2, L_2 \rangle \) at literal \( L \) iff there exists a sub-argument of \( \langle A_2, L_2 \rangle, \langle A, L \rangle, \langle A_1, L_1 \rangle \) such that \( \langle A_1, L_1 \rangle \) counterargues \( \langle A_2, L_2 \rangle \) at \( L \) and neither \( \langle A_1, L_1 \rangle \) nor \( \langle A, L \rangle \) preferred over \( \langle A, L \rangle \) preferred over \( \langle A, L \rangle \) preferred over \( \langle A, L \rangle \). If \( \langle A_1, L_1 \rangle \) is either a proper or a blocking defeater of \( \langle A_2, L_2 \rangle \), it is said to be a defeater of the latter.

An argumentation line for an argument \( \langle A_0, L_0 \rangle \) is a sequence \( \lambda = [ \langle A_0, L_0 \rangle, \langle A_1, L_1 \rangle, \langle A_2, L_2 \rangle, \ldots ] \) where for each \( i > 0 \) \( \langle A_{i+1}, L_{i+1} \rangle \) is a defeater of \( \langle A_i, L_i \rangle \), \( \lambda_S = [ \langle A_0, L_0 \rangle, \langle A_2, L_2 \rangle, \langle A_4, L_4 \rangle, \ldots ] \) is the sequence of supporting argument of \( \lambda \) while the sequence of interfering ones is \( \lambda_I = [ \langle A_1, L_1 \rangle, \langle A_3, L_3 \rangle, \langle A_5, L_5 \rangle, \ldots ] \).

An acceptable argumentation line in a defeasible program \( P = (\Pi, \Delta) \) is a finite sequence \( \lambda = [ \langle A_0, L_0 \rangle, \ldots, \langle A_n, L_n \rangle ] \) such that some constraints on the addition of arguments to the sequence are considered. For example, not allowing the addition of an already introduced argument, or using an argument simultaneously as a supporting and as an interfering argument (see García and Simari 2004 for an in-depth analysis of the problem). A constraint that we will postulate for any acceptable argumentation line to hold is concordance of all supporting respectively interfering arguments.

Definition 3 (concordance) A set of arguments \( A_i, 1 \leq i \leq m, \) of a defeasible logic program \( (\Pi, \Delta) \) is called concordant iff \( \forall i \in \Pi \cup \bigcup_{j=1}^m A_i \) is non-contradictory.

To answer a query \( Q \), the warrant procedure builds up a candidate argument \( \langle A, Q \rangle \). Then, it associates to this argument a dialectical tree \( T_{(A,Q)} \) as follows:

1. The root of the tree is labeled, \( \langle A_0, Q_0 \rangle, \) i.e., \( A_0 = A \) and \( Q_0 = Q \).
2. Let \( n \) be a non-root node, with label \( \langle A_n, Q_n \rangle \) and \( \lambda = [ \langle A_0, Q_0 \rangle, \ldots, \langle A_n, Q_n \rangle ] \) the labels in the path from the root to \( n \). Let \( \mathcal{B} = \{ (B_1, R_1), \ldots, (B_k, R_k) \} \) be the set of all the defeaters for \( \langle A_n, Q_n \rangle \). For \( 1 \leq i \leq k \), if \( \lambda = [ \langle A_0, Q_0 \rangle, \ldots, \langle A_n, Q_n \rangle, (B_i, R_i) ] \) is an acceptable argumentation line, \( n \) has a child \( n_i \) labeled \( (B_i, R_i) \). If \( \mathcal{B} = \emptyset \) or no \( (B_i, R_i) \in \mathcal{B} \) is such that \( \lambda \) is acceptable, then \( n \) is a leaf of the tree.

The nodes of \( T_{(A,Q)} \) can be marked \( U \) (undefeated) or \( D \) (defeated), yielding a tagged tree \( T^{*}_{(A,Q)} \) as follows:

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

Then, given an argument \( \langle A, Q \rangle \) and its associated tagged tree \( T^{*}_{(A,Q)} \), if the root is marked \( U \), the literal \( Q \) is said to be warranted, and \( A \) is said to be the warrant for \( Q \).

In this paper, we will be assuming that there are no strict rules, so that the strict logical knowledge is represented by facts, and will write \( \Phi \) instead of \( \Pi \). With this prerequisite, argument \( \langle A_1, L_1 \rangle \) attacks another \( \langle A_2, L_2 \rangle \) at a literal \( L \) if there exists a sub-argument \( \langle A, L \rangle, \langle A_2, L_2 \rangle \) such that \( L = \overline{L} \). Concordance can be checked easily, too.

**Proposition 4** Let \( P = (\Pi, \Delta) \) be a defeasible logic program the strict part \( \Pi \) of which contains only facts. A set of arguments \( A_1, \ldots, A_m \) of \( P \) is concordant iff for any two rules \( \delta_1 \in A_i, \delta_2 \in A_j \), it holds that head(\( \delta_1 \)) \neq \text{head}(\delta_2) \).

Defeasible rules can be taken as conditionals, and we will use both terms synonymously. In the rest of the paper, a defeasible logic program \( P = (\Phi, \Delta) \) consists of a collection of facts, \( \Phi \), and conditionals, \( \Delta \), both specified in a suitable propositional language \( L \) with \( L \) being the apportioning set of possible worlds. As a further important prerequisite, we postulate that the set \( \Delta \) of conditionals be consistent (as defined in section 2). Then, this setting is fully compatible to the methodology of OCF’s and system Z described in section 2, and we will elaborate this connection in the following sections. For ease of notation, \( \Phi \) will also denote the conjunction of all facts of \( P \).

3 A semantics of examples and counterexamples for arguments

The basic idea of argumentation is to support beliefs by arguments that are often built from rules. This logical underpinning of arguments is crucial for DELP, but can not be taken into regard by Dung’s abstract argumentation framework (Dung 1995) which is considered to be one of the standard approaches to give semantics to arguments. Here, we propose a semantical approach to arguments which works particularly well if the arguments have a logical structure but might be applied more generally. Our approach differs from Dung’s framework and also from more logical approaches like those presented in (Besnard and Hunter 2008). The basic idea first to be developed in this paper is that arguments are as convincing as their (most plausible) examples. We will introduce the notion of examples and counterexamples of arguments in this section to justify a priority relation between arguments. Plausibility issues will be taken into consideration in the next section.

Possible worlds verifying all conditionals occurring in an argument may be used as examples to illustrate the statement of this argument.
For the arguments $A_1$, $L_1$, $A_2$, $L_2$, $A_3$, $f$ stated in example 2, examples and counterexamples are given as follows:

\[
\begin{align*}
(A_1, f)^+ &= \text{Mod}(\text{csb}f) \\
(A_2, \overline{f})^+ &= \text{Mod}(\text{cs}f) \\
(A_3, f)^+ &= \text{Mod}(\text{cs}f)
\end{align*}
\]

Hence, $\omega_1 = \text{csb}f w$ is an example of $(A_1, f)$ and $(A_3, f)$ and a counterexample of $(A_2, \overline{f})$. Reciprocally, $\omega_2 = \text{cs}f w$ is an example of $(A_2, \overline{f})$, and a counterexample of $(A_1, f)$ and $(A_3, f)$.

Relationships between arguments can be characterized easily by considering their examples:

\[\text{as usual, we set } \land \emptyset = \top \text{ and } \lor \emptyset = \bot.\]
4 Priorities based on plausibility

To decide whether an argument not only attacks but also defeats another, we need a priority relation among arguments to provide a proper criterion. Here, we will base this priority relation on degrees of plausibility that can be assigned to examples of arguments by ordinal conditional functions. We consider an argument to be as plausible as its most plausible examples. Moreover, the plausibilities of its counterexamples represent the degree to which it can be challenged.

Definition 13 (κ-values of arguments) Let κ be an ordinal conditional function on Ω, let ⟨A, L⟩ be an argument. Then κ+(⟨A, L⟩) = min{κ(ω) | ω ∈ ⟨A, L⟩+}, and κ−(⟨A, L⟩) = min{κ(ω) | ω ∈ ⟨A, L⟩−}.

When comparing two arguments, the argument the example of which is more plausible prevails.

Definition 14 (κ-preference) Let κ be an ordinal conditional function on Ω, let ⟨A1, L1⟩, ⟨A2, L2⟩ be two arguments. Then ⟨A1, L1⟩ ⪰ κ ⟨A2, L2⟩ iff κ(⟨A1+⟩) ≤ κ(⟨A2+⟩).

By using κ-preference as priority relation between DELP-arguments, an easy criterion for warrant can be derived.

Proposition 15 Let ⟨A, L⟩ be an argument. If κ+(⟨A, L⟩) < κ−(⟨A, L⟩), then ⟨A, L⟩ is undefeated and hence warranted.

Proof: Let κ+(⟨A, L⟩) < κ−(⟨A, L⟩). Assume there is an argument ⟨A1, L1⟩ that (attacks and) defeats ⟨A, L⟩. Then, according to proposition 9 and definition 14, there must be an example ω1 ∈ ⟨A1, L1⟩+ that is a counterexample to ⟨A, L⟩ such that κ(ω1) ≤ κ+(⟨A, L⟩). But this contradicts the presupposition κ+(⟨A, L⟩) < κ−(⟨A, L⟩). □

Therefore, making use of arbitrary κ-functions allows for a straightforward and intuitive semantical specification of priority between DELP-arguments. In the following section, we will focus on a specific κ-function – namely system Z – to develop a more syntactical priority relation.

5 Using system Z preference for DELP

In the original DELP-framework (García and Simari 2004), generalized specificity was used to define a priority relation between arguments, taking the derivation behaviour of arguments into account. In system Z (Goldszmidt and Pearl 1996) and section 2), a tolerance partitioning of the set of defeaters is used to define a priority relation between arguments. In this paper, we make use of the Z-ordering on the conditionals in Δ for this purpose and will be able to sharpen proposition 15.

Since we presupposed Δ to be consistent, there is a tolerance partitioning Δ = Δ0 ∪ ... ∪ Δk, i.e. the one used by system Z (cf. section 2). Then, we can measure the plausibility of an argument ⟨A, L⟩ by the plausibility of its most plausible rule by setting

\[ Z(⟨A, L⟩) = \min_{δ ∈ A} Z(δ). \]

Note that in fact, the most plausible rules are the weakest links in the chain supporting the conclusion of an argument. Now, arguments can be compared by their Z-values.


To check if a literal is warranted by an argument we have to build up the dialectical tree for this argument from acceptable argumentation lines. Using Z-preference, an argumentation line [⟨A0, L0⟩, ⟨A1, L1⟩, ⟨A2, L2⟩, ...] is acceptable iff all supporting arguments [⟨A0, L0⟩, ⟨A2, L2⟩, ⟨A3, L3⟩, ...], respectively interfering arguments [⟨A1, L1⟩, ⟨A3, L3⟩, ⟨A5, L5⟩, ...] are concordant, and each blocking defeater is followed by a proper defeater, i.e. ⟨A_m+1, L_{m+1}⟩ ≥Z ⟨A_m, L_m⟩ and ⟨A_{m+1}, L_{m+1}⟩ >Z ⟨A_{m-1}, L_{m-1}⟩ must hold.

We will check which literals can be warranted in our penguin & chicken example 1.

Example 17 We consider the delp P1 = (⟨cs⟩, Δ), with propositional and conditionals as specified in example 1. The Z-values of the arguments ⟨A1, f⟩, ⟨A2, f⟩ and ⟨A3, f⟩ as given in example 2 are computed as Z(⟨A1, f⟩) = 0, Z(⟨A2, f⟩) = 1, Z(⟨A3, f⟩) = 2, so ⟨A2, f⟩ properly defeats ⟨A1, f⟩, and ⟨A3, f⟩ properly defeats ⟨A2, f⟩. Hence, ⟨A1, f⟩ is finally undefeated in the dialectical tree which is shown in figure 1, so f is warranted.

In the example above, the warrant procedure proved to be in accordance with system Z. In the next example, argumentation is shown to do even better than system Z.

Example 18 We work again in the setting of example 1, now considering the defeasible logic program P2 = (⟨p⟩, Δ). We will check if w is warranted, i.e. if penguins have wings can be proved in our argumentation framework. The only argument that can be built to connect p and w is ⟨(⟨b⟩p), (w|b)⟩, which is not attacked at all, so, in particular, is undefeated. Hence, w can be warranted.

On the other hand, in system Z, we have κ_z(pw) = κ_z(pw|b) = 1, since each of the models to be considered falsifies at least one of (⟨b⟩p), (f|b), or (f|b), the most plausible of which is (f|b) – is in Δ0, just as (w|b). So, the status of the query w can not be determined by system Z. This effect has become known as the drowning effect (see, e.g. (Goldszmidt and Pearl 1996)).
We will now prove some general results on arguments and their examples. The following lemma is straightforward, taking into regard that subarguments are actually subsets of their superarguments.

**Lemma 19** Let \( \langle A_1, L_1 \rangle \) be a subargument of \( \langle A_2, L_2 \rangle \). Then \( Z(A_1) \geq Z(A_2) \).

Next, we elaborate a relationship between the Z-value of an argument and its counterexamples.

**Lemma 20** Let \( \langle A, L \rangle \) be an argument. Then for all counterexamples \( \omega \in \langle A, L \rangle^{-} \), \( \kappa_{\omega}(\omega) \geq Z(A) + 1 \).

**Proof:** Each \( \omega \in \langle A, L \rangle^{-} \) falsifies at least one rule \( \delta \) in \( A \) with rank \( Z(\delta) \geq Z(A) \), so \( \kappa_{\omega}(\omega) \geq Z(A) + 1 \).

Finally, we show how the warrant of an argument can be ensured by “good” examples in this framework.

**Proposition 21** Let \( \langle A, L \rangle \) be an argument. If \( \kappa_{\omega}(\langle A, L \rangle) \leq Z(\langle A, L \rangle) \), then \( \langle A, L \rangle \) is undefeated, hence warranted.

**Proof:** Assume there is \( \langle A_1, L_1 \rangle \) that defeats \( \langle A, L \rangle \) at subargument \( \langle A', L' \rangle \). Then \( Z(\langle A_1, L_1 \rangle) \geq Z(\langle A', L' \rangle) \geq Z(\langle A, L \rangle) + 1 \). Each example \( \omega \) of \( \langle A', L' \rangle \) falsifies at least one rule in \( \langle A_1, L_1 \rangle \) so \( \kappa_{\omega}(\omega) \geq Z(\langle A_1, L_1 \rangle) + 1 \geq Z(\langle A, L \rangle) + 1 \) for all \( \omega \in \langle A', L' \rangle^{-} \). Since \( \langle A, L \rangle^+ \subseteq \langle A', L' \rangle^+ \), by Proposition 12, and \( \langle A, L \rangle^+ = Mod(\Phi \land \bigwedge_{\delta \in A} \text{head}(\delta)) \), by Proposition 6, this implies \( \kappa_{\omega}(\bigwedge_{\delta \in A} \text{head}(\delta)) \geq Z(\langle A, L \rangle) \), a contradiction. □

The following corollary is a straightforward consequence.

**Corollary 22** Let \( \langle A, L \rangle \) be an argument. If there is an example \( \omega \) of \( \langle A, L \rangle \) such that \( \kappa_{\omega}(\omega) \leq Z(\langle A, L \rangle) \), then \( \langle A, L \rangle \) is undefeated, hence warranted.

**Example 23** We apply these last results to our penguin & chicken example. For \( \langle A_3, f \rangle \), we compute \( \kappa_{\omega}(\langle A_3, f \rangle) = \kappa_{\omega}(csbf) = 2 \) and \( Z(\langle A_3, f \rangle) = 2 \), so from proposition 21, we can conclude that \( f \) is warranted by \( \langle A_3, f \rangle \), identifying \( \langle A_3, f \rangle \) immediately as a “good” argument for \( f \). A “good” example for \( f \) in the sense of corollary 22 can be given by \( \omega_1 = cshbf \). Note that \( \langle A_1, f \rangle \) can not be established as a “good” argument for \( f \) by proposition 21, since \( \kappa_{\omega}(\langle A_1, f \rangle) = \kappa_{\omega}(csbf) = 2 \), whereas \( Z(\langle A_1, f \rangle) = 0 \).

From the last part of example 23, it is obvious that proposition 21 states a sufficient but not necessary condition for warrant in the DeLP-framework.

### 6 Conclusion

The novel combination of defeasible argumentation with methods from default and conditional reasoning elaborated in this paper was shown to be fruitful in several respects: First, a possible worlds semantics widely used in default reasoning provides a novel semantics for DeLP arguments by taking possible worlds as examples. This allows for attaching a notion of plausibility to arguments in a straightforward way, thus being able to associate a meta-logical priority criterion to the DeLP-framework. Finally, we considered the particular system Z approach to plausibility and proved that in some cases, the complex dialectical evaluation procedure of DeLP can be circumvented by directly comparing the Z-values of examples to the Z-values of the arguments. In this way, each of the two frameworks enriches the other one by new methods and insights. It must be emphasized that DeLP features a fully dialectical frame for argumentation more expressive than other approaches to argumentative inference. In (Benferhat, Dubois, and Prade 1993), argumentative inference is based only on comparing degrees of possibility so that attack there is the same as defeat whereas DeLP also uses the notion of defense to warrant arguments.

Further work might continue in different directions: For example, the Z-value of arguments here was based on the minimum of the Z-degrees of the involved rules, taking the most general (i.e. weakest) rules as a reference point. Other choices are possible, one might go for the most specific rules here, or compare the conflicting rules directly. The aim to be kept in mind would be to specify the results of DeLP’s dialectical inference procedure most appropriately in a declarative way. On the other hand, with a focus on default reasoning, one might investigate in more depth in which cases system Z inferences can be realized as results of argumentation processes. Either way, although default reasoning and argumentation are both techniques usable for commonsense reasoning, our impression is that these fields rather complement one another in many fruitful ways than compete with each other.

### References


