Artificial Intelligence 53 (1992) 125–157
Elsevier

# A mathematical treatment of defeasible reasoning and its implementation

Guillermo R. Simari and Ronald P. Loui

*Department of Computer Science, Washington University in Saint Louis,*
*Campus Box 1045, One Brookings Drive, Saint Louis, MO 63130-4899, USA*

Received February 1990
Revised April 1991

*Abstract*

Simari, G.R. and R.P. Loui, A mathematical treatment of defeasible reasoning and its implementation, Artificial Intelligence 53 (1992) 125–157.

We present a mathematical approach to defeasible reasoning based on arguments. This approach integrates the notion of specificity introduced by Poole and the theory of warrant presented by Pollock. The main contribution of this paper is a precise, well-defined system which exhibits correct behavior when applied to the benchmark examples in the literature. It aims for usability rather than novelty.

We prove that an order relation can be introduced among equivalence classes of arguments under the equi-specificity relation. We also prove a theorem that ensures the termination of the process of finding the justified facts. Two more lemmas define a reduced search space for checking specificity.

In order to implement the theoretical ideas, the language is restricted to Horn clauses for the evidential context. The language used to represent defeasible rules has been restricted in a similar way.

The authors intend this work to unify the various existing approaches to argument-based defeasible reasoning.

## 1. Introduction

Recent courage to deviate from standard practice in nonmonotonic reasoning has led to an influx of formalisms. Each achieves nonmonotonicity in a first-order language where entailment is not based on fixed points, nor on model minimization. Most avoid intensional contexts by semantic ascent,[1] thus supplementing the proof theory in the metalanguage. This obviates the need for model-theoretic accounts of new syntax, since there is no new syntax.

[1] Quine's phrase, in private communication.

Inspiration has come from conditional logics (Nute [9, 10], Delgrande [1], Glymour and Thomason [3]) or inductive logics. In the latter case, both induction's form (Loui [7], Pollock [12]) and its effect (Geffner and Pearl [2], Neufeld [8]) have been copied. All of the resulting systems incorporate a specificity defeater, analogous to the subclass defeater in inheritance with exceptions (since Touretzky [20]).

Some of these authors have found use for objects called *arguments* (also *theories*). Other systems are based on irrelevance. This paper is concerned with those based on arguments.

Arguments are *prima facie* proofs that may make use of assertions that one sentence is (defeasible) reason for another. They indicate support for a proposition, but do not establish warrant once and for all; it matters what other counterarguments there may be. Arguments may have stucture (Loui [7], Pollock [12]) or may just be collections of supporting sentences (Poole [14], Geffner and Pearl [2]). There is widespread agreement that arguments in these systems generalize paths in inheritance systems.[2]

As is the case in inheritance, there is a "clash of intuitions" that has resulted in a plethora of theories. There are at present few ways of classifying the systems. Our intent, in defining yet another system, is not to add to the inventory. In fact, this paper attempts to bring together the prominent systems based on arguments. A system is defined that takes its form from Loui (which in turn evolved from that of Kyburg [6]) and which combines the rules of Poole and of Pollock. For most of the AI audience, this will effectively condense three systems into one, remedying deficiencies of each.

More importantly, this system is defined in a mathematically more rigorous manner. Past definitions (especially Poole's and Loui's) did not have the precision nor the completeness to serve as a foundation for future mathematical work. It is no accident that the statement of the system here allows concise proof of nontrivial properties.

### 1.1. Poole and Pollock combined

Poole treats specificity, i.e., a comparative measure of the relevance of information, in an elegant and usable way, but does not describe adequately when to apply his *specificity comparator* to interactions among arguments. On the other hand, Pollock treats the interaction among arguments properly while rejecting specificity. Pollock rejects specificity both as a generalization of the subclass defeater and as a useful shorthand. This places him in an extreme minority in the defeasible reasoning community.

In our view, Poole and Pollock fail to develop the best ideas in their systems to produce a system of lasting usefulness to the knowledge representation

[2] General discussion during the Workshop on Defeasible Reasoning with Specificity and Multiple Inheritance, St. Louis, MO (1989).

community. Poole [13] has implemented a system of defeasible reasoning which does not address interactions among arguments. Pollock has taken his research in a direction which is too general for AI's uses.

The system defined here combines the ideas of the two. But the main contribution of this paper is a precise, well-defined system which exhibits a correct behavior when applied to the benchmark examples in the literature.[3]

We take the knowledge of agent $\boldsymbol{a}$ to be divided into a set of defeasible rules $\Delta$ and a set of well-formed formulas (*wffs*) $\mathcal{K}$ in the standard formal logic sense. The set $\mathcal{K}$ is further divided into (1) grounded wffs: the *contingent* part of $\mathcal{K}$; and (2) ungrounded wffs, the *necessary* part of $\mathcal{K}$. Evidence suggests new tentative conclusions; a potential conclusion $\boldsymbol{p}$ will be suggested when it is consistent with $\mathcal{K}$ and has a supporting subset of $\Delta$ which, in conjunction with $\mathcal{K}$, can derive $\boldsymbol{p}$ without deriving a contradiction at the same time.

Accepting or rejecting $\boldsymbol{p}$ is a matter of comparing arguments supporting $\boldsymbol{p}$, their counterarguments, their rebuttals, and so on. $\boldsymbol{p}$ must be consistent with $\mathcal{K}$, but its interaction with subsets of $\Delta$ could be more interesting. If a subset $S$ of $\Delta$ supports $\boldsymbol{p}$, we will say that there exists a defeasible derivation of $\boldsymbol{p}$ from $S$. The subsets form *argument structures*, and are ordered according to Poole.

Poole claims his specificity relation is based on Popperian ideas, but some find it unintuitive or lacking justification. We view it as a convention: arguably, the most useful convention to date. It is based on the four-part observation that

(1) two conflicting arguments were made;
(2) sometimes one argument can be made while the other cannot;
(3) the reverse is not true;
(4) thus, one argument is more particular about the current evidence than the other; it is more specific.

Extrapolating from the total evidence requirement of inductive logic, being more particular about the evidence makes an argument stronger. Another way to rationalize the rule is pragmatic: if the more specific argument does not defeat the less specific, then it is never an effective argument, since the less specific argument can always be made as a counterargument whenever the first argument can be made. This is unacceptable for representing knowledge.

Pollock's method of defining which arguments survive counterargument and actually justify their conclusions is appealing. Essentially, it propagates defeat from arguments that have no defeaters, and it could be defined in any of a number of ways. We retain Pollock's original inductive step to recognize his contribution, but the rule could be expressed as TMS-like labeling, or as AND–OR graph evaluation. We have two kinds of labels while Pollock has only one: this is a purely technical variation on Pollock that we introduce

[3] Actual solution of two dozen such examples can be obtained from the authors.

because defeat is implicit in this theory, while it is explicit in Pollock's. It is the implicit defeat arising from comparison of specificity that makes the hybrid system attractive for actual use.

## 2. Arguments and specificity

We will construct a formal system $\mathbb{L}$ with the objective of providing a language in which to represent the knowledge of a given agent $\boldsymbol{a}$ and in which to perform defeasible reasoning.

The language of $\mathbb{L}$ is composed of a *first-order language* $\mathscr{L}$, plus a binary metalinguistic relation among members of $\mathscr{L}$. Any axiomatization of $\mathscr{L}$ will do for our purposes, and we will use the standard *connectives* and *punctuation symbols* freely without explicitly introducing them. We assume that the rules of inference attached to the axiomatization are *modus ponens* and *generalization*. The members of the metalinguistic relation are called *defeasible rules* and they have the form $\alpha \mathrel{>\!\!-} \beta$, where $\alpha$ and $\beta$ are well-formed formulas (*wffs*) in $\mathscr{L}$, which must contain free variables, e.g., they are nonclosed wffs. The relation "$\mathrel{>\!\!-}$" among $\mathscr{L}$'s wffs is understood as expressing that "reasons to believe in the antecedent $\alpha$ provide reasons to believe in the consequent $\beta$". Variables with the same name on both sides of the rule are assumed to be the same. An instance of an open defeasible rule is obtained by replacing *all* the free variables by appropriate constants. When no confusion is possible we will use the term defeasible rule to refer to the open defeasible rule and to its grounded instances.

The set *Sent*$(\mathscr{L})$ of *sentences* of $\mathscr{L}$, that is the set of closed well-formed formulas in $\mathscr{L}$, can be partitioned in two subsets, corresponding to necessary and contingent information. Necessary information is the context in which defeasible rules are provided. Although a purely syntactic distinction might not be possible on philosophical grounds, we normally take sentences with variables or implication to be necessary. Thus, the first subset contains the grounded sentences $Sent_{\mathrm{C}}(\mathscr{L})$ and the second subset contains nongrounded sentences $Sent_{\mathrm{N}}(\mathscr{L})$, i.e.,

$$Sent(\mathscr{L}) = Sent_{\mathrm{C}}(\mathscr{L}) \cup Sent_{\mathrm{N}}(\mathscr{L})\,.$$

Obviously,

$$Sent_{\mathrm{C}}(\mathscr{L}) \cap Sent_{\mathscr{L}}(\mathscr{L}) = \emptyset\,.$$

The names used for the subsets reflect the view that the grounded sentences in $Sent_{\mathrm{C}}(\mathscr{L})$ represent information depending on the individual constants of the language. Those individual constants are *contingent* to the reality being represented. On the other hand, the sentences in $Sent_{\mathrm{N}}(\mathscr{L})$ are wffs containing variables. That characteristic allows them to convey properties that single out a *class* of worlds, i.e., worlds where the relations among individuals are the same regardless of the *local* individuals. We choose to call these sentences the *necessary* facts, because without them the world would not be as it is.

The knowledge of $\boldsymbol{a}$ is represented in $\mathbb{L}$ by a pair $(\mathcal{K}, \Delta)$, where $\mathcal{K}$ is a subset of $Sent(\mathcal{L})$, and $\Delta$ is a finite set of defeasible rules. The pair $(\mathcal{K}, \Delta)$ will be called a *defeasible logic structure*. $\mathcal{K}$ represents the indefeasible part of $\boldsymbol{a}$'s knowledge and $\Delta$ represents tentative information, i.e., information that $\boldsymbol{a}$ is prepared to take at less than face value. In mapping $\boldsymbol{a}$'s reality to a subset $\mathcal{K}$ of $\mathcal{L}$ we obtain a partition of $\mathcal{K}$ in two subsets

$$\mathcal{K}_{\mathrm{N}} = Sent_{\mathrm{N}}(\mathcal{L}) \cap \mathcal{K}\,, \qquad \mathcal{K}_{\mathrm{C}} = Sent_{\mathrm{C}}(\mathcal{L}) \cap \mathcal{K}\,.$$

Clearly, $\mathcal{K} = \mathcal{K}_{\mathrm{N}} \cup \mathcal{K}_{\mathrm{C}}$. The only condition on $\mathcal{K}$ is consistency, i.e., $\mathcal{K} \not\vdash \bot$. Sometimes, when using $\mathcal{K}$, we will refer to it as the *context*, and $\mathcal{K}$ will be considered as a set of wffs or as the conjunction of them depending on the situation.

Having defined our knowledge representation language we need to introduce a notion of entailment, or inference, which is somewhat different from the one used in first-order languages. That is, given a defeasible logic structure $(\mathcal{K}, \Delta)$, we need to define what other facts can be sanctioned as *justified*. Our formal system introduces this notion in a way that is not axiomatic. For a complete definition we need further develop our formalism. We will present the syntactic part here. The rest will be introduced in the next sections.

Given a member $A$ of $Sent(\mathcal{L})$, and set $\Gamma = \{A_1, A_2, \ldots, A_n\}$, where each $A_i$ is a member of $\mathcal{K}$ or a grounded instance of a member of $\Delta$, we will establish a meta-meta-relationship "$\mathrel{|\!\!\sim}$", called *defeasible consequence*, between $\Gamma$ and $A$ in the following way. A well-formed formula $A$ will be called a defeasible consequence of the set $\Gamma$ as described above, if and only if there exists a sequence $B_1, \ldots, B_m$ such that $A = B_m$ and, for each $i$, either $B_i$ is an axiom of $\mathcal{L}$ or $B_i$ is in $\Gamma$, or $B_i$ is a direct consequence of the preceding members of the sequence using modus ponens or instantiation of a universally quantified sentence. The grounded instances of the defeasible rules are regarded as material implications for the application of modus ponens.[4] The sequence $B_1, \ldots, B_m$ will be called a *defeasible derivation* or just *a derivation*. We use $\Gamma \mathrel{|\!\!\sim} A$ as an abbreviation of $A$ *is a defeasible consequence of* $\Gamma$. If necessary, in order to avoid confusion with the context, we write $\Gamma \mathrel{|\!\!\sim}_{\mathcal{K}} A$. We also will write $A_1, \ldots, A_n \mathrel{|\!\!\sim} A$ instead of $\{A_1, \ldots, A_n\} \mathrel{|\!\!\sim} A$, and $\mathcal{K} \cup T \mathrel{|\!\!\sim} A$, making explicit the distinction between the context $\mathcal{K}$ and a set $T$ of defeasible rules used in the derivation.

In first-order logic the above definition is enough to describe the wffs that are theorems of $\Gamma$, but for us the situation is different because we need to

[4] Since modus ponens is unidirectional, this does not imply reasoning by modus tollendo ponens, or contrapositive reasoning. In fact, the latter two are not allowed in this system.

introduce the tentative nature of the conclusions, e.g., we need to give a criterion that will allow us to prefer one conclusion over another. That criterion will be the specificity relation among arguments. We will now introduce the formal notion of arguments and later we will define the specificity relation among those formal objects.

### 2.1. Arguments

Derivations, as defined above, make use of some grounded instances of defeasible rules from $\Delta$. The set of grounded defeasible rules characterize the derivation and we will give the name *argument basis* to that set. In order to facilitate the following discussion we introduce the set $\Delta^{\downarrow}$ of all grounded instances of members of $\Delta$ produced by using the individual constants in $\mathscr{L}$.

**Definition 2.1** (*Preliminary*). Given a context $\mathscr{K} = \mathscr{K}_{\mathrm{N}} \cup \mathscr{K}_{\mathrm{C}}$ and a set $\Delta$ of defeasible rules we say that a subset $T$ of $\Delta^{\downarrow}$, is an *argument* for $h \in Sent_{\mathrm{C}}(\mathscr{L})$ in the context $\mathscr{K}$, denoted by $\langle T, h \rangle_{\mathscr{K}}$, if and only if:

(1) $\mathscr{K} \cup T \mid\!\sim h$,

(2) $\mathscr{K} \cup T \mid\!\not\sim \bot$.

The pair $\langle T, h \rangle_{\mathscr{K}}$ will be called an *argument structure*.

**Remark 2.2.** When possible we will drop the reference to the context and we will write $\langle T, h \rangle$ meaning $\langle T, h \rangle_{\mathscr{K}}$. We will refer to the collection of all possible argument structures as **AStruc($\Delta^{\downarrow}$)** or just **AStruc**. There is a distinguished argument, $\langle \emptyset, \mathscr{K}^{\vdash} \rangle$, for any context $\mathscr{K}$ with finitely representable closure; i.e., no rules are necessary to support the conjunction of the atoms of the deductive closure ($\mathscr{K}^{\vdash}$) of the knowledge in $\mathscr{K}$. Finally, for $\langle T, h \rangle$ we will assume that the set $T$ is minimal, or nonredundant in the sense that it does not contain any rule that is unnecessary for the inference of $h$. This is a sort of "Occam's razor" principle for arguments.

**Definition 2.3** (*Revised*). Given a context $\mathscr{K} = \mathscr{K}_{\mathrm{N}} \cup \mathscr{K}_{\mathrm{C}}$ and a set $\Delta$ of defeasible rules we say that a subset $T$ of $\Delta^{\downarrow}$, is an *argument* for $h \in Sent_{\mathrm{C}}(\mathscr{L})$ in the context $\mathscr{K}$, denoted by $\langle T, h \rangle_{\mathscr{K}}$, if and only if:

(1) $\mathscr{K} \cup T \mid\!\sim h$,

(2) $\mathscr{K} \cup T \mid\!\not\sim \bot$,

(3) $\nexists T' \subset T, \mathscr{K} \cup T' \mid\!\sim h$.

**Example 2.4.** Let $\mathscr{K} = \{P(a), Q(a)\}$ and

$$\Delta = \{P(x) \succ\!\!- R(x),\ Q(x) \vee R(x) \succ\!\!- H(x),\ M(x) \succ\!\!- N(x)\}$$

be the context and defeasible rule set respectively. Therefore the subset $T$ of grounded instances of defeasible rules in $\Delta$,

$$T = \{P(a) \succ\!\!- R(a), Q(a) \wedge R(a) \succ\!\!- H(a)\} ,$$

is an argument structure for $H(a)$, i.e. $\langle T, H(a)\rangle$ is an argument structure.

**Definition 2.5.** Let $\langle T, h\rangle$ be an argument structure for $h$, and $\langle S, j\rangle$ an argument structure for $j$ such that $S \subseteq T$. We will say that $\langle S, j\rangle$ is a *subargument of* $\langle T, h\rangle$ and use the notation $\langle S, j\rangle \subseteq \langle T, h\rangle$, overloading the symbol "$\subseteq$".

**Example 2.6.** Given any argument structure $\langle T, h\rangle$, the two argument structures $\langle \emptyset, \mathcal{K}^{\vdash}\rangle$ and $\langle T, h\rangle$ are two trivial subarguments of it.

**Example 2.7.** In the conditions of the above example,

$$S_1 = \{P(a) \succ\!\!- R(a), Q(a) \wedge R(a) \succ\!\!- H(a)\}$$

is an argument for $H(a)$, and

$$S_2 = \{P(a) \succ\!\!- R(a)\}$$

is an argument for $R(a)$. We have the following relations among the argument structures: $\langle S_1, H(a)\rangle \subseteq \langle S_1, H(a)\rangle$ and $\langle S_2, R(a)\rangle \subseteq \langle S_1, H(a)\rangle$.

Sometimes it will be necessary to talk about the defeasible rules in terms of their antecedents and consequents. The following definitions introduce three operators for this purpose.

**Definition 2.8.** Let $T$ be a finite subset of $\Delta^{\downarrow}$. We will introduce two operators over sets of defeasible rules. They are the operator $An(\cdot)$, which applied to $T$ will return the set of antecedents of its rules, and $Co(\cdot)$, which applied to $T$ will return the set of consequents of its rules. $Sent_{\mathrm{C}}(\mathscr{L})$ is normally restricted to conjunctions from $An(\Delta^{\downarrow})$.

**Example 2.9.** Given the argument

$$T = \{A(r) \succ\!\!- D(r), B(r) \wedge D(r) \succ\!\!- C(r), C(r) \succ\!\!- E(r)\} ,$$

we have

$$An(T) = \{A(r), B(r), D(r)\} , \qquad Co(T) = \{D(r), C(r), E(r)\} .$$

It is also useful to have access to the set of literals used in the defeasible rules of an argument structure.

**Definition 2.10.** Let $\langle T, h\rangle$ be an argument structure. Then the operator $Lit(\cdot)$ will return the set of literals in $T$ with the exception of $h$, i.e., $Lit(\langle T, h\rangle) = An(T) \cup Co(T) - \{h\}$.

**Example 2.11.** Given the argument $T$ as in Example 2.9, we have $Lit(\langle T, E(r)\rangle) = \{A(r), B(r), C(r), D(r)\}$.

### 2.2. *Specificity*

Having defined these objects we would like to establish certain binary relations on **AStruc**($\Delta^{\downarrow}$) in such a way that it would help us to choose the "better" argument structure that supports a conclusion. The following definitions, essentially Poole's [14], will characterize this relation.

**Definition 2.12.** Given two argument structures $\langle T_1, h_1\rangle$ and $\langle T_2, h_2\rangle$ in **AStruc**, we say that $T_1$ for $h_1$ is *strictly more specific than* $T_2$ for $h_2$ denoted by

$$\langle T_1, h_1\rangle \succ_{\text{spec}} \langle T_2, h_2\rangle ,$$

if and only if:

(1) $\forall e \in Sent_C(\mathcal{L})$ such that $\mathcal{K}_N \cup \{e\} \cup T_1 \mathrel{|\!\!\sim} h_1$ and $\mathcal{K}_N \cup \{e\} \not\mathrel{|\!\!\sim} h_1$ also $\mathcal{K}_N \cup \{e\} \cup T_2 \mathrel{|\!\!\sim} h_2$, and

(2) $\exists e \in Sent_C(\mathcal{L})$ such that:

$$\mathcal{K}_N \cup \{e\} \cup T_2 \mathrel{|\!\!\sim} h_2 \quad \text{(activates } T_2\text{)} ,$$

$$\mathcal{K}_N \cup \{e\} \cup T_1 \not\mathrel{|\!\!\sim} h_1 \quad \text{(does not activate } T_1\text{)} ,$$

$$\mathcal{K}_N \cup \{e\} \not\mathrel{|\!\!\sim} h_2 \quad \text{(nontriviality condition)} .$$

**Remark 2.13.** The term *activates* appearing in the definition is used with the following meaning: *together with $\mathcal{K}_N$ the argument $T$ is enough to construct a defeasible derivation of $h$.*

Another important relation among argument structures is the notion of being equally specific.

**Definition 2.14.** Two argument structures $T_1$ for $h_1$ and $T_2$ for $h_2$ are *equi-specific*, denoted by

$$\langle T_1, h_1\rangle \approx_{\text{spec}} \langle T_2, h_2\rangle ,$$

when the following condition holds,

$$\forall e \in Sent_C(\mathcal{L}) ,$$

$$\mathcal{K}_N \cup \{e\} \cup T_1 \mathrel{|\!\!\sim} h_1 \text{ if and only if } \mathcal{K}_N \cup \{e\} \cup T_2 \mathrel{|\!\!\sim} h_2 .$$

Finally the combination of both notions gives the following definition.

**Definition 2.15.** We say that an argument structure $T_1$ for $h_1$ is *at least as specific as* an argument structure $T_2$ for $h_2$ denoted by

$$\langle T_1, h_1 \rangle \geq_{\text{spec}} \langle T_2, h_2 \rangle \,,$$

if and only if $\langle T_2, h_2 \rangle \approx_{\text{spec}} \langle T_1, h_1 \rangle$ or $\langle T_1, h_1 \rangle >_{\text{spec}} \langle T_2, h_2 \rangle$.

Some examples will clarify the concept.

**Example 2.16.** The argument structure $\langle \{A(r) \wedge B(r) >\!\!- C(r)\}, C(r) \rangle$ is more specific than $\langle \{A(r) >\!\!- \neg C(r)\}, \neg C(r) \rangle$ because every time the first argument can be activated to support $C(r)$ the second also supports $\neg C$. But, on the other hand, $A(r)$ alone can activate the second argument structure but does not activate the first. So

$$\langle \{A(r) \wedge B(r) >\!\!- C(r)\}, C(r) \rangle >_{\text{spec}} \langle \{A(r) >\!\!- \neg C(r), \neg C(r) \rangle \,.$$

**Example 2.17.** The argument structure $\langle \{A(r) >\!\!- \neg C(r)\}, \neg C(r) \rangle$ is more specific than the argument structure $\langle \{A(r) >\!\!- B(r), B(r) >\!\!- C(r)\}, C(r) \rangle$ because every time the first argument can support $\neg C(r)$ the second also supports $C(r)$. But, on the other hand, $B(r)$ alone can activate the second argument structure but does not activate the first. So

$$\langle \{A(r) >\!\!- \neg C(r)\}, \neg C(r) \rangle >_{\text{spec}} \langle \{A(r) >\!\!- B(r), B(r) >\!\!- C(r)\}, C(r) \rangle \,.$$

**Remark 2.18.** Whenever no confusion is possible we will drop the subscript "spec" in the symbols "$>_{\text{spec}}$", "$\geq_{\text{spec}}$", and "$\approx_{\text{spec}}$" writing instead "$>$", "$\geq$" and "$\approx$".

An argument and its subarguments are related by the specificity relation in a natural, expected way.

**Lemma 2.19.** *Let $\langle T, h \rangle$ be an argument structure and $\langle S, j \rangle$ a subargument of $\langle T, h \rangle$. Then $\langle T, h \rangle$ is more specific than $\langle S, j \rangle$, i.e., $\langle T, h \rangle \geq \langle S, j \rangle$.*

The equi-specificity "$\approx$" relation decomposes **AStruc** into disjunct subsets of equi-specific arguments, i.e., establishes a partition on it. This porperty is better expressed in the following lemma.

**Lemma 2.20.** *The equi-specificity relation among members of* **AStruc** *is an equivalence relation.*

The "$\approx$" equivalence relation will help us to introduce an order relation in the set **AStruc**$/\approx$, i.e., the quotient set of **AStruc** by the equivalence relation "$\approx$". This order relation is induced by "$\geq$". First, we observe that $\geq$ defines a quasi-ordering in **AStruc**, i.e., the relation is reflexive and transitive. If we lift the relation to the quotient set **AStruc**$/\approx$ of the equivalence classes defined by

"$\approx$" in **AStruc** the new relation will define a partial order over those classes, as is shown in the lemma below.

**Remark 2.21.** For all $\langle T, h\rangle$ in **AStruc** the notation $[\langle T, h\rangle]$ represents the equivalence class of $\langle T, h\rangle$ in **AStruc**$/\approx$.

**Definition 2.22.** We define the relation "$\sqsupseteq_{\text{spec}}$" in the quotient set **AStruc**$/\approx$ as follows. Given $[\langle T_1, h_1\rangle]$ and $[\langle T_2, h_2\rangle]$ in **AStruc**$/\approx$,

$$[\langle T_1, h_1\rangle] \sqsupseteq [\langle T_2, h_2\rangle] \text{ if and only if } \langle T_1, h_1\rangle \succeq \langle T_2, h_2\rangle\,.$$

Again, whenever possible we will drop the "spec" subscript from "$\sqsupseteq_{\text{spec}}$" writing "$\sqsupseteq$".[5]

The introduction of the $\succeq$ relation of **AStruc** has the objective of providing a way to select the most "appropriate" argument structure. In that sense the following lemma establishes the fundamental property regarding order in **AStruc**$/\approx$.

**Lemma 2.23.** *The relation $\sqsupseteq$ defined in* **AStruc**$/\approx$ *is a partial order.*

The next lemma defines a reduced search space for checking specificity.

**Lemma 2.24.*** *Let $\langle T_1, h_1\rangle$ and $\langle T_2, h_2\rangle$ be two argument structures in* **AStruc**. *Then the following conditions are equivalent*:

(1) $\langle T_1, h_1\rangle \succeq \langle T_2, h_2\rangle$,

(2) $\forall x \in An(T_2)$, $\mathcal{K}_N \cup An(T_1) \cup T_2 \mathrel{|\!\!\sim} x$.

**Proof.**

(1) implies (2). Assume that $\langle T_1, h_1\rangle \succeq \langle T_2, h_2\rangle$. Hence, $\langle T_1, h_1\rangle$ is at least as specific as any subargument of $\langle T_2, h_2\rangle$. There is always a subargument, $S$, of $\langle T_2, h_2\rangle$ for any $x$ in $An(T_2)$ (by the nonredundant property of $\langle T_2, h_2\rangle$). $\langle T_1, h_1\rangle \succeq \langle S, x\rangle$. Since $An(T_1)$ activates $T_1$ for $h_1$, it activates $S$ for $x$. Therefore, $\mathcal{K}_N \cup An(T_1) \cup T_2 \mathrel{|\!\!\sim} x$ for all $x \in An(T_2)$.

(2) implies (1). Assume that $e$ in $Sent_C(\mathcal{L})$ is such that $\mathcal{K}_N \cup \{e\} \cup T_1 \mathrel{|\!\!\sim} h_1$. We want to show that $\mathcal{K}_N \cup \{e\} \cup T_2 \mathrel{|\!\!\sim} h_2$. Because of the quantification in (2), every $An(T_2)$ can be derived, therefore every $Co(T_2)$ is defeasibly derived; hence $\langle T_2, h_2\rangle$ is activated, i.e., $\mathcal{K}_N \cup \{e\} \cup T_2 \mathrel{|\!\!\sim} h_2$. That is, $\langle T_1, h_1\rangle \succeq \langle T_2, h_2\rangle$. □

**Lemma 2.25.** *Let $\langle T_1, h_1\rangle$ and $\langle T_2, h_2\rangle$ be such that $\langle T_1, h_1\rangle \succeq \{T_2, h_2\rangle$. Let*

[5] As noted by a referee, we do not state the theorem that given $\langle T_1, h_1\rangle \approx \langle T_2, h_2\rangle$ and $\langle T_1, h_1\rangle \succeq \langle T_3, h_3\rangle$, also $\langle T_2, h_2\rangle \succeq \langle T_3, h_3\rangle$, but this is immediate from the definitions.

* At press time, this lemma and its proof are found to be in error. A corrigendum is planned by the authors.

*$\langle T_2, h_2 \rangle$ be such that $\forall x \in Co(T_2)$, $\mathcal{K} \cup T_1 \mid\!\sim x$. If $\langle T_2, h_2 \rangle$ contains a subargument structure $\langle R, p \rangle$, then $\langle T_1, h_1 \rangle$ contains a subargument structure $\langle S, p \rangle$ such that $\langle S, p \rangle \succeq \langle R, p \rangle$.*

**Proof.** The subargument structure $\langle R, p \rangle$ of $\langle T_2, h_2 \rangle$ is formed by the subset $R$ of $T_2$. Given that every member in $Co(T_2)$ can be inferred using the rules in $T_1$ and $\mathcal{K}$, we can distinguish which rules are necessary to prove the subset $Co(R)$ of $Co(T_2)$, calling it $S$. We contend that $\langle S, p \rangle$ is the required subargument. Obviously, for all $x$ in $Co(R)$, $\mathcal{K} \cup S \mid\!\sim x$, by its own definition. Therefore, any literal necessary to infer $p$ from $R$ is available in $S$. For the same reason $\langle S, p \rangle \succeq \langle R, p \rangle$. □

This establishes conditions for discarding arguments which reduces the search for argument defeaters.

**Remark 2.26.** Given two arguments $\langle T_1, h_1 \rangle$ and $\langle T_2, h_2 \rangle$ satisfying the conditions of the above lemma, we will say that $\langle T_1, h_1 \rangle$ *covers* $\langle T_2, h_2 \rangle$.

## 3. An algebra of arguments

A very good question regarding arguments is about the kind of operations that it is possible to define on them. We will devote the next few sections to consider certain operations on $\mathcal{F}(\langle T, h \rangle) = \{\langle T_i, h_i \rangle\}_{i \in I}$ the family of subarguments of an arbitrary argument structure $\langle T, h \rangle \in$ **AStruc**, where $I$ is a set of indices, and explore some of its properties and interrelations. When no confusion is possible we will use $\mathcal{F}$ instead of $\mathcal{F}(\langle T, h \rangle)$.

A set of wffs in a first-order language is consistent if and only if there is no formula for which that formula and its negation are theorems of the set. Our defeasible derivation relation is weaker than derivation in first-order logic. It is possible to defeasibly derive contradictory conclusions from an arbitrary set of defeasible rules. Because of that characteristic we will introduce a weaker notion for sets of defeasible rules.

The following discussion omits proofs which can be found in the thesis [17].

**Definition 3.1.** Given two argument structures $\langle T_1, h_1 \rangle$ and $\langle T_2, h_2 \rangle$ in **AStruc** they will be called *concordant* if $\mathcal{K} \cup T_1 \cup T_2 \not\mid\!\sim \perp$.

As could be expected, subarguments of a given argument structure have the property of being concordant with each other.

**Proposition 3.2.** *Let $\langle T, h \rangle \in$ **AStruc** be an arbitrary argument structure, and let $\mathcal{F}$ be the family $\{\langle T_i, h_i \rangle\}_{i \in I}$ of all $\langle T, h \rangle$ subargument structures, then the members of $\mathcal{F}$ are pairwise concordant.*

### 3.1. *Argument combination (join)*

**Definition 3.3.** Let $\langle T, h\rangle \in$ **AStruc** be an arbitrary argument structure, and let $\mathcal{F}$ be the family of all its subargument structures. Given $\langle T_1, h_1\rangle$ and $\langle T_2, h_2\rangle$ in $\mathcal{F}$, we define the *combination* of them as the argument structure $\langle T_3, h_3\rangle$, where $T_3 = T_1 \cup T_2$ and $h_3 = h_1 \wedge h_2$. The operation will be denoted:

$$\langle T_3, h_3\rangle = \langle T_1, h_1\rangle \sqcup \langle T_2, h_2\rangle .$$

**Proposition 3.4.** *The combination of argument structures is a well-defined operation in $\mathcal{F}$.*

**Proposition 3.5.** *Given two argument structures $\langle T_1, h_1\rangle$ and $\langle T_2, h_2\rangle \in \mathcal{F}$, the combination $\langle T_3, h_3\rangle = \langle T_1, h_1\rangle \sqcup \langle T_2, h_2\rangle$ is such that $\langle T_3, h_3\rangle \geq \langle T_1, h_1\rangle$ and $\langle T_3, h_3\rangle \geq \langle T_2, h_2\rangle$ and $\langle T_3, h_3\rangle$ is the minimal (in $\geq$) argument structure in $\mathcal{F}$ with that property which contains $\langle T_1, h_1\rangle$ and $\langle T_2, h_2\rangle$ as subarguments.*

**Proposition 3.6** (Associativity). *The combination of arguments in $\mathcal{F}$ is associative, i.e., if $\langle T_1, h_1\rangle$, $\langle T_2, h_2\rangle$, and $\langle T_3, h_3\rangle$ are subargument structures of $\mathcal{F}$ then*

$$(\langle T_1, h_1\rangle \sqcup \langle T_2, h_2\rangle) \sqcup \langle T_3, h_3\rangle$$
$$= \langle T_1, h_1\rangle \sqcup (\langle T_2, h_2\rangle \sqcup \langle T_3, h_3\rangle) .$$

**Proposition 3.7** (Commutativity). *The combination of arguments in $\mathcal{F}$ is commutative, i.e., if $\langle T_1, h_1\rangle$, $\langle T_2, h_2\rangle$ are subargument structures of $\mathcal{F}$ then*

$$\langle T_1, h_1\rangle \sqcup \langle T_2, h_2\rangle = \langle T_2, h_2\rangle \sqcup \langle T_1, h_1\rangle .$$

**Definition 3.8.** Given a subfamily $\{\langle T_{i_j}, h_{i_j}\rangle\}_{i_j \in J}$ of $\{(T_i, h_i\rangle\}_{i \in I}$, we define the *generalized combination* of the subargument structures in it as

$$\bigsqcup_{j \in J} \{\langle T_{i_j}, h_{i_j}\rangle\}_{i_j \in J} = \left\langle \bigcup_{j \in J} T_{i_j}, \bigwedge_{j \in J} h_{i_j} \right\rangle .$$

**Proposition 3.9.** *The argument structure $\langle \emptyset, \mathcal{K}^{\vdash}\rangle$ (if $\mathcal{K}^{\vdash}$ is finitely representable) is an identity element with respect to the combining operation in the family $\mathcal{F}$ of subargument structures of a given argument structure $\langle T, h\rangle$.*

### 3.2. *Argument intersection (meet)*

Given a subset $T$ of $\Delta^{\downarrow}$, we will describe the rules on it as $\{A_i \succ\!\!- B_i\}_{i \in I}$. Using that representation we can consider the set $\{A_i\}_{i \in I}$ of antecedents of rules in $T$ and the set $\{B_i\}_{i \in I}$ of consequents of those rules. If $\langle T, h\rangle$ is an argument structure for $h$, then the set $(\mathcal{K} \cup \{B_i\}_{i \in I})^{\vdash}$ is the set of literals for which there is a subargument structure contained in $\langle T, h \rangle$.

**Definition 3.10.** A set of rules $\{A_i \succ\!\!- B_i\}_{i \in I}$ is *consistent* if and only if $\{B_i\}_{i \in I} \nvdash \bot$.

**Remark 3.11.** For arguments $\langle T, h \rangle$, $T$ is consistent because of the nonredundancy and the $\mid\!\sim$-consistency of arguments.

Let $T$ be an arbitrary, but consistent, subset of $\Delta^{\downarrow}$. The question is "Is there a literal in $\mathcal{L}$ for which we can have an argument structure using $T$?" The literal $(\mathcal{K} \cup \{B_i\}_{i \in I})^{\vdash}$ has that property. It also has the property of being the strongest literal, in the usual sense, with that property.

**Definition 3.12.** Let $\langle T, h \rangle \in$ **AStruc** be an arbitrary argument structure, and let $\mathcal{F}$ be the family of all its subargument structures. Given $\langle T_1, h_1 \rangle$ and $\langle T_2, h_2 \rangle$ in $\mathcal{F}$, we define the *intersection* of them as the argument structure $\langle T_3, h_3 \rangle$, where $T_3 = T_1 \cap T_2$ and $h_3$ is defined as $(\mathcal{K} \cup \{B_i\}_{i \in I})^{\vdash}$, where $\{B_i\}_{i \in I}$ is the set of consequents of the rules in $T_3$. The operation will be denoted:

$$\langle T_3, h_3 \rangle = \langle T_1, h_1 \rangle \sqcap \{T_2, h_2 \rangle \, .$$

**Proposition 3.13.** *The intersection of argument structures is a well-defined operation in* $\mathcal{F}$.

**Proposition 3.14.** *Given two argument structures* $\langle T_1, h_1 \rangle$ *and* $\langle T_2, h_2 \rangle \in \mathcal{F}$, *the intersection* $\langle T_3, h_3 \rangle = \langle T_1, h_1 \rangle \sqcap \{T_2, h_2 \rangle$ *is such that* $\langle T_1, h_1 \rangle \succeq \langle T_3, h_3 \rangle$ *and* $\langle T_2, h_2 \rangle \succeq \langle T_3, h_3 \rangle$ *and* $\langle T_3, h_3 \rangle$ *is the maximal argument structure in* $\mathcal{F}$ *with the property of being a subargument of* $\langle T_1, h_1 \rangle$ *and* $\langle T_2, h_2 \rangle$.

**Proposition 3.15** (Associativity). *The intersection of arguments in* $\mathcal{F}$ *is associative, i.e., if* $\langle T_1, h_1 \rangle$, $\langle T_2, h_2 \rangle$, *and* $\langle T_3, h_3 \rangle$ *are subargument structures in* $\mathcal{F}$ *then*

$$\begin{aligned}
&(\langle T_1, h_1 \rangle \sqcap \langle T_2, h_2 \rangle) \sqcap \langle T_3, h_3 \rangle \\
&\quad = \langle T_1, h_1 \rangle \sqcap (\langle T_2, h_2 \rangle \sqcap \langle T_3, h_3 \rangle) \, .
\end{aligned}$$

**Proposition 3.16** (Commutativity). *The intersection of arguments in* $\mathcal{F}$ *is commutative, i.e., if* $\langle T_1, h_1 \rangle$, $\langle T_2, h_2 \rangle$ *are subargument structures in* $\mathcal{F}$ *then*

$$\langle T_1, h_1 \rangle \sqcap \langle T_2, h_2 \rangle = \langle T_2, h_2 \rangle \sqcap \langle T_1, h_1 \rangle \, .$$

**Definition 3.17.** Given a subfamily $\{\langle T_{i_j}, h_{i_j} \rangle\}_{i_j \in J}$ of $\{\langle T_i, h_i \rangle\}_{i \in I}$, we define the generalized intersection of the subargument structures in it as:

$$\bigsqcap_{j\in J}\{\langle T_{i_j}, h_{i_j}\rangle\}_{i_j\in J}=\left\langle\bigcap_{j\in J} T_{i_j}, (\mathscr{K}\cup\{B_i\}_{i\in I})^{\vdash}\right\rangle ,$$

where $\{B_i\}_{i\in I}$ is the set of consequents in $\cap_{j\in J} T_{i_j}$.

**Proposition 3.18.** *The family $\mathscr{F}$ of subargument structures of a given argument structure $\langle T, h\rangle$ has an identity element with respect to the intersection operation. That identity element is the argument structure $\langle T, h\rangle$.*

***Corollary* 3.19.** *The family $\mathscr{F}$ with the intersection and combination operations defined over the argument structures forms a lattice.*

## 4. Justifications

In the previous section we introduced the notion of argument structure and defined a specificity relationship on the set of all possible argument structures. The reason to define that relationship is to be able to "select" argument structures with the characteristic of being "better" than others. In this section we will define the selection process.

### *4.1. Basic interactions among argument structures*

Arguments are objects that represent "pieces" of the reasoning process. They relate to each other in different ways. We have already seen an example in the *subargument* relation. Another example is the *concordance* among argument structures, i.e., the property which would allow to join them without producing an inconsistency. Going in the opposite direction is the *disagreement* relation that will be introduced in the next subsection. Some other interactions involving specificity are possible. We will introduce them now starting with those that are simplest to define.

*4.1.1. Disagreement*

It is possible for two argument structures to support two facts which together with the context $\mathscr{K}$ are inconsistent. We will refer to the relationship between two argument structures in that situation as *disagreement*. Next we will present the formal definition of disagreement.

**Definition 4.1.** We say that two argument structures $T_1$ for $h_1$ and $T_2$ for $h_2$

*disagree*, denoted by

$$\langle T_1, h_1\rangle \bowtie_{\mathcal{H}} \langle T_2, h_2\rangle$$

if and only if $\mathcal{H} \cup \{h_1, h_2\} \vdash \bot$.

The following are examples of this relationship.

**Example 4.2.**

$$\langle\{E >\!\!- \neg C\}, \neg C\rangle \bowtie_{\mathcal{H}} \langle\{A \wedge B >\!\!- C\}, C\rangle\,, \quad \mathcal{H} = \{E, A, B\}\,.$$

**Example 4.3.**

$$\langle\{E >\!\!- \neg C\}, \neg C\rangle \bowtie_{\mathcal{H}} \langle\{A >\!\!- X\}, X\rangle\,, \quad \mathcal{H} = \{E, A, X \supset C\}\,.$$

The following is not an example of this relationship, but motivates the next definition.

**Example 4.4.**

$$\langle\{E >\!\!- \neg B\}, \neg B\rangle, \langle\{E >\!\!- B, B >\!\!- A\}, A\rangle\,, \quad \mathcal{H} = \{E\}\,.$$

### *4.1.2. Counterargument*

The counterargument relation tells us about the internal construction of an argument structure with reference to another argument structure. It is a refinement of the disagreement relation. It looks to the subarguments of a given argument structure in light of another argument, i.e., indicates the existence of subarguments of an argument structure which are in disagreement with the other argument. Formally:

**Definition 4.5.** We say that an argument structure $T_1$ for $h_1$ *counterargues* an argument structure $T_2$ for $h_2$ at $h$, denoted by

$$\langle T_1, h_1\rangle \otimes\!\!\rightarrow^{h} \langle T_2, h_2\rangle$$

if and only if there exists a subargument $\langle T, h\rangle$ of $\langle T_2, h_2\rangle$ such that $\langle T_1, h_1\rangle \bowtie_{\mathcal{H}} \langle T, h\rangle$, i.e., $\langle T, h\rangle \subset \langle T_2, h_2\rangle$ and $\mathcal{H} \cup \{h_1, h\} \vdash \bot$.

**Remark 4.6.** A fact $h$ in the conditions of Definition 4.5 will be referred to as a *counterargument point.*

**Example 4.7.**

$$\langle\{E >\!\!- \neg C\}, \neg C\rangle \otimes\!\!\rightarrow^{C} \langle\{A \wedge B >\!\!- C, C >\!\!- D\}, D\rangle\,,$$

where $\langle\{E >\!\!- \neg C\}, \neg C\rangle$ is in disagreement with the subargument $\langle\{A \wedge B >\!\!- C\}, C\rangle$ of $\langle\{A \wedge B >\!\!- C, C >\!\!- D\}, D\rangle$.

#### 4.1.3. Defeat

The *defeat* relationship is a further refinement of counterargument, where the specificity relation comes into play. We will say that an argument structure $\langle T_1, h_1\rangle$ *defeats* another argument structure $\langle T_2, h_2\rangle$ if it is the case that $\langle T_2, h_2\rangle$ contains a subargument structure $\langle T, h\rangle$ such that $\langle T_1, h_1\rangle$ disagrees with $\langle T, h\rangle$, and $\langle T_1, h_1\rangle$ is more specific than $\langle T, h\rangle$. That is:

**Definition 4.8.** We say that an argument structure $T_1$ for $h_1$ *defeats* an argument structure $T_2$ for $h_2$, denoted by

$$\langle T_1, h_1\rangle \gg_{\text{def}} \langle T_2, h_2\rangle$$

if and only if there exists a subargument structure $\langle T, h\rangle$ of $\langle T_2, h_2\rangle$ such that:

(1) $\langle T_1, h_1\rangle \otimes\!\!\rightarrow^h \langle T_2, h_2\rangle$, i.e., $T_1$ for $h_1$ counterargues $T_2$ for $h_2$ at $h$, and

(2) $\langle T_1, h_1\rangle \succ \langle T, h\rangle$, i.e., $T_1$ for $h_1$ is more specific than $T$ for $h$.

**Remark 4.9.** A fact $h$ in the conditions of Definition 4.8 will be referred to as a *defeater point*.

**Example 4.10.**

$$\langle\{A \wedge B \wedge E \succ\!\!- \neg C\}, \neg C\rangle \gg_{\text{def}} \langle\{A \wedge B \succ\!\!- C, C \succ\!\!- D\}, D\rangle\,,$$

that is, the argument structure $\langle\{A \wedge B \wedge E \succ\!\!- \neg C\}, \neg C\rangle$ counterargues $\langle\{A \wedge B \succ\!\!- C, C \succ\!\!- D\}, D\rangle$ at $C$ and $\langle\{A \wedge B \wedge E \succ\!\!- \neg C\}, \neg C\rangle$ is more specific than $\langle\{A \wedge B \succ\!\!- C\}, C\rangle$.

### 4.2. Justifying arguments

A fundamental issue in reasoning is to decide what the agent believes as a function of a given context and the set of defeasible rules forming his explicit knowledge. But how can he decide if a tentative conclusion is part of the implicit knowledge? Or how can he decide if that tentative conclusion is consistent with the implicit knowledge? According to our scheme this decision must be taken by analyzing what kind of support the tentative conclusion has. This can be accomplished by seeing which arguments are relevant to the conclusion.

Given a fact $h$, there may be several argument structures in the set **AStruc**($\Delta^{\downarrow}$) of argument structures formed with members of $\Delta^{\downarrow}$, which *support* $h$ from the context $\mathcal{K}$. Those argument structures relate to others in **AStruc**($\Delta^{\downarrow}$) by the defeat and counterargument relations. For an argument structure $\langle T, h\rangle$ in **AStruc**($\Delta^{\downarrow}$), we may have a set $I$ of argument structures which *interfere* with $\langle T, h\rangle$, i.e., they counterargue $\langle T, h\rangle$. In $I$, the set of interfering arguments, there may be some arguments which defeat $\langle T, h\rangle$.

Those *defeaters* could in turn be defeated. If all the defeaters are defeated, the original argument structure $\langle T, h\rangle$ becomes reinstated. The above discussion leads to an inductive definition, which is similar to Pollock's [12] and characterizes that process.

**Definition 4.11.** Arguments are active at various levels as supporting or interfering arguments.

(1) All arguments are (level-0) S-arguments (supporting arguments) and I-arguments (interfering arguments).

(2) An argument $\langle T_1, h_1\rangle$ is a (level-$(n+1)$) S-argument if and only if there is no level-$n$ I-argument $\langle T_2, h_2\rangle$ such that for some $h$, $\langle T_2, h_2\rangle$ counterargues $\langle T_1, h_1\rangle$ at $h$, i.e., $\nexists \langle T_2, h_2\rangle \in \mathbf{AStruc}$ such that, for some $h$, $\langle T_2, h_2\rangle \otimes\!\!\rightarrow^h \langle T_1, h_1\rangle$.

(3) An argument $\langle T_1, h_1\rangle$ is a (level-$(n+1)$) I-argument if and only if there is no level-$n$ I-argument $\langle T_2, h_2\rangle$ such that $\langle T_2, h_2\rangle$ defeats $\langle T_1, h_1\rangle$.

**Remark 4.12.** A level-$n$ S-argument will be denoted by $\mathrm{S}^n$-argument and a level-$n$ I-argument will be denoted by $\mathrm{I}^n$-argument. Also notice that we dropped the parentheses.

**Definition 4.13.** We say that an argument $\langle T, h\rangle$ in **AStruc** *justifies* $h$ if and only if there exists $m$ such that, for all $n \geqslant m$, $\langle T, h\rangle$ is an $\mathrm{S}^n$-argument for $h$. We say that $h$ is *justified in* $\Omega \subseteq$ **AStruc** if there is a $\langle T, h\rangle \in \Omega$ that justifies $h$.

**Lemma 4.14.** *Let $\langle T, h\rangle$ be an argument structure in* **AStruc**, *such that $\langle T, h\rangle$ justifies $h$. Then every subargument $\langle R, q\rangle$ of $\langle T, h\rangle$ justifies its conclusion $q$.*

**Proof.** The proof comes from the fact that any possible defeater of $\langle R, q\rangle$ will also be a defeater for $\langle T, h\rangle$. And since $\langle T, h\rangle$ justifies $h$, no effective defeater exists. □

We say that $h$ is *provisionally justified* at level $n$ iff there exists an $\mathrm{S}^n$-argument which supports it. A set of provisionally justified facts is called *stable* iff every member of it is justified.

It is possible to define a sequence $\{\Sigma^n\}$ of operators over **AStruc** in correspondence with Definition 4.11 in the following way. For a given $k$, let $\Sigma^k$(**AStruc**) be the set of $h$ such that there exists $\langle T, h\rangle$ that is in **AStruc** and is an $\mathrm{S}^k$-argument; i.e., $\Sigma^k$ produces the set of partially justified facts at level $k$. This definition allows us to talk about the set of justified facts in operational terms, as in the following lemmas.

**Lemma 4.15.** *If* $\Sigma^n$(**AStruc**) $= \Sigma^{n+1}$(**AStruc**), *then* $\Sigma^n$(**AStruc**) *is stable.*

**Proof.** The proof of this lemma is obvious from the definition of stable set. Once $\Sigma$ has "repeated" itself, i.e., $\Sigma^n(\mathbf{AStruc}) = \Sigma^{n+1}(\mathbf{AStruc})$, that means that no new interfering argument has been reinstated. Therefore, no $\mathrm{I}^n$-argument can get defeated at level $n + 1$ and no $\mathrm{S}^n$-argument can get counterargued. □

Now the open question is whether that situation is ever reached. The next theorem will answer that question.

**Theorem 4.16.** *For any defeasible logic structure* $(\mathcal{K}, \Delta)$ *with finite* **Astruc**, *there is a unique stable set, and the operator* $\Sigma$ *will find it.*

**Proof** (Sketch). The set $\mathbf{AStruc}/{\approx}$, as we have shown previously is partially ordered by "$\sqsupseteq$". Consider the set $\mathcal{P}(\mathbf{AStruc}/{\approx})$ of all the subsets of $\mathbf{AStruc}/{\approx}$. Some of their members are totally ordered sets, i.e., chains. These chains are formed by equivalence classes which contain equi-specific arguments. But even though two arguments are equi-specific they may support different facts. Nevertheless, from one of the chains in $\mathcal{P}(\mathbf{AStruc}/{\approx})$ we can extract chains of arguments which support the same fact. Being finite, from those chains it is possible to extract the most specific argument for every fact. We collect all the most specific arguments in a set that for reference convenience we will call $\mathcal{T}(\mathbf{AStruc})$. Notice here that we may have in $\mathcal{T}(\mathbf{AStruc})$ more of one argument for a fact, but if that case occurs the argument structures are unrelated by "$\succeq$".

We apply the justification procedure of Definition 4.13 to $\mathcal{T}(\mathbf{AStruc})$, and this is equivalent to applying it to **AStruc**, as is clear from the following discussion. It is obvious that we have lost no interesting argument by restricting ourselves to $\mathcal{T}(\mathbf{AStruc})$. For every argument structure in **AStruc**, there is an argument structure in $\mathcal{T}(\mathbf{AStruc})$ which is at least as specific as the one in **AStruc**. So in looking for counterarguments of an argument structure we will obtain the same counterargument points. The same is true for defeaters, with the difference that now we only have to look at the more specific argument structures possible for a defeater point.

Now, given one of these arguments in $\mathcal{T}(\mathbf{AStruc})$, $\langle T, h\rangle$, we consider the set *Counter*$(\langle T, h\rangle)$ of counterarguments of $\langle T, h\rangle$ in $\mathcal{T}(\mathbf{AStruc})$. Obviously the set can be empty. The set *Counter*$(\langle T, h\rangle)$ contains only the more specific counterarguments for every possible counterargument point of $\langle T, h\rangle$. For every member $\langle R, q\rangle$ in *Counter*$(\langle T, h\rangle)$ there is a set of possible defeaters *Defeaters*$(\langle R, q\rangle)$ which contains only defeater arguments which are more specific than $\langle R, q\rangle$; again the set can be empty. *Defeaters* $(\langle R, q\rangle)$ contains only the more specific defeaters $\langle S, r\rangle$ for every possible defeater point of $\langle R, q\rangle$. This construction can be performed until we get a "tree" where the nodes of the tree are connected by the "counterargument" relation, the root to its children, or the "defeat" relation between the rest of the levels. This tree contains the whole dialectical structure for the argument being considered.

We define an *argument line* as the walk that it is possible to construct from one of the leaf nodes of the tree to the first node before the root, i.e., the last node in an argument line is a counterargument. We can apply the second operation defined in the justification procedure to the set of argument lines obtained from the tree. If an argument line "survives" the test the argument is defeated.

These argument lines are sequences $\langle\langle T_1, h_1\rangle, \langle T_2, h_2\rangle, \ldots, \langle T_n, h_n\rangle\rangle$ where $\langle T_1, h_1\rangle$ is the counterargument. The sequence is ordered by the specificity relation, i.e., $\langle T_n, h_n\rangle \succ \cdots \succ \langle T_2, h_2\rangle \succ \langle T_1, h_1\rangle$. If $n$ is odd, the counterargument survives; if $n$ is even, the counterargument is defeated as is easy to see. □

**Definition 4.17.** We will refer to the stable set defined by Theorem 4.16 as $\Sigma_\infty$.

## 5. Discarding arguments

In this section we will show some relationships among arguments and justifications aiming to find avenues pointing to efficient implementations. In that direction, it is importa~~n~~t to find properties that will characterize arguments that can be discarded in order to reduce the size of the search space. Again, proofs are omitted.

We prove propositions essential to proving the claim that covered arguments can be discarded. This is a weak pruning method, and serious investigation of pruning will have to look at stronger claims. But our interest here is how the formalism allows provable claims.

**Lemma 5.1.** *Given two argument structures $\langle T_1, h_1\rangle$ and $\langle T_2, h_2\rangle$ in* **AStruc** *such that $\langle T_1, h_1\rangle \succeq \langle T_2, h_2\rangle$ and $\mathcal{K} \cup \{h_1\} \vdash \{h_2\}$, then $\langle T_1, h_2\rangle$ is an argument structure. That is, $T_1$ is an argument for $h_2$, and $\langle T_1, h_2\rangle \succeq \langle T_2, h_2\rangle$, i.e. $T_1$ for $h_2$ is more specific than $T_2$ for $h_2$.*

**Lemma 5.2.** *Given two argument structures $\langle T_1, h_1\rangle$ and $\langle T_2, h_2\rangle$ such that $\langle T_1, h_1\rangle \succeq \langle T_2, h_2\rangle$ and $\mathcal{K} \cup \{h_1\} \vdash \{h_2\}$, if $h_2$ is justified in $\Omega \cup \{\langle T_1, h_1\rangle, \langle T_2, h_2\rangle\}$ then $h_2$ is justified in $\Omega \cup \{\langle T_1, h_1\rangle\}$, where $\Omega$ is any subset of* **AStruc**.

**Lemma 5.3.** *Given two argument structures $\langle T_1, h_1\rangle$ and $\langle T_2, h_2\rangle$ such that $\langle T_1, h_1\rangle$ covers $\langle T_2, h_2\rangle$, i.e., $\langle T_1, h_1\rangle \succeq \langle T_2, h_2\rangle$ and $\mathcal{K} \cup T_1 \mathrel{|\!\!\sim} x$, for all $x$ in $Co(T_2)$, then if $p$ is justified in $\Omega \cup \{\langle T_1, h_1\rangle, \langle T_2, h_2\rangle\}$, $p$ is also justified in $\Omega \cup \{\langle T_1, h_1\rangle\}$, where $\Omega$ is any subset of* **AStruc**.

**Proposition 5.4.** *Given $\langle T_1, h_1\rangle$, $\langle T_2, h_2\rangle$, and $\langle T, h\rangle$ in* **AStruc**, *where*

*$\langle T_1, h_1\rangle$ covers $\langle T_2, h_2\rangle$, then if $\langle T_2, h_2\rangle$ contains a subargument structure $\langle R, p\rangle$ such that $\langle R, p\rangle \otimes\!\!\rightarrow^{p} \langle T, h\rangle$, then $\langle T_1, h_1\rangle$ contains a subargument structure $\langle S, p\rangle$ such that $\langle S, p\rangle \otimes\!\!\rightarrow^{p} \langle T, h\rangle$.*

This allows covered arguments to be discarded while keeping the arguments that cover them, with no loss in ability to counterargue.

**Proposition 5.5.** *Given $\langle T_1, h_1\rangle$, $\langle T_2, h_2\rangle$, and $\langle T, h\rangle$ in* **AStruc** *such that $\langle T_1, h_1\rangle$ covers $\langle T_2, h_2\rangle$, then if $\langle T_2, h_2\rangle \gg_{\mathrm{def}} \langle T, h\rangle$ then $\langle T_1, h_1\rangle \gg_{\mathrm{def}} \langle T, h\rangle$.*

This allows covered arguments to be discarded while keeping the arguments that cover them, with no loss in ability to defeat.

Given $\Omega \subseteq$ **AStruc** and $\langle T_1, h_1\rangle, \langle T_2, h_2\rangle, \langle T, h\rangle \in$ **AStruc**, where $\langle T_1, h_1\rangle$ covers $\langle T_2, h_2\rangle$, define

$$\Omega_{\mathrm{big}} = \Omega \cup \{\langle T, h\rangle, \langle T_1, h_1\rangle, \langle T_2, h_2\rangle\}\,,$$

$$\Omega_{\mathrm{small}} = \Omega \cup \{\langle T, h\rangle, \langle T_1, h_1\rangle\}\,.$$

**Proposition 5.6.** *If $\langle T, h\rangle$ is an $S^n$-argument in $\Omega_{\mathrm{big}}$, then $\langle T, h\rangle$ is an $S^n$-argument in $\Omega_{\mathrm{small}}$.*

This is the inductive step toward completing the argument that covered arguments can be discarded if at least one of their covers is retained. The supporting arguments are not disrupted by discarding a covered argument.

## 6. Some interesting examples

We will show some examples presented in the literature of defeasible reasoning to show the behavior of the system.

**Example 6.1** (*Opus does not fly*). An example of how information regarding a subclass overrides more general information corresponding to superclasses.

| | |
|---|---|
| Birds tend to fly | $(B(x) >\!\!- F(x))$ , |
| Penguins tend to not fly | $(P(x) >\!\!- \neg F(x))$ , |
| All penguins are birds | $(P(x) \supset B(x))$ , |
| Opus is a Penguin | $(P(opus))$ , |
| Does Opus fly? | $(F(opus)?)$ . |

The context and defeasible rule set are

$$\mathcal{K} = \{P(opus),\ P(opus) \supset B(opus)\}\,,$$

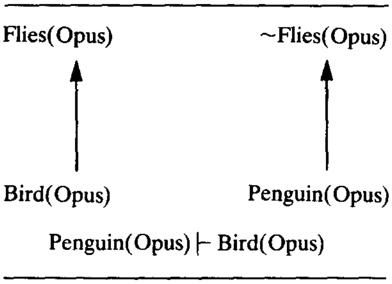


Fig. 1. Example 6.1.

$$\Delta = \{B(opus) \succ\!\!- F(opus), P(opus) \succ\!\!- \neg F(opus)\}$$

respectively (see Fig. 1). Two argument structures are interesting:

$$\langle T_1, F(opus)\rangle = \langle \{B(opus) \succ\!\!- F(opus)\}, F(opus)\rangle ,$$

$$\langle T_2, \neg F(opus)\rangle = \langle \{P(opus) \succ\!\!- \neg F(opus)\}, \neg F(opus)\rangle .$$

We have the following disagreement

$$\langle T_1, F(opus)\rangle \bowtie_{\mathcal{K}} \langle T_2, \neg F(opus)\rangle .$$

Moreover

$$\langle T_2, \neg F(opus)\rangle \otimes\!\!\rightarrow^{\neg F(opus)} \langle T_1, F(opus)\rangle .$$

But

$$\langle T_2, \neg F(opus)\rangle > \langle T_1, F(opus)\rangle .$$

Therefore,

$$\langle T_2, \neg F(opus)\rangle \geqslant_{\text{def}} \langle T_1, F(opus)\rangle ,$$

hence $\langle T_2, \neg F(opus)\rangle$ justifies $\neg F(opus)$.

**Example 6.2** (*Nixon Diamond*). This canonical example is devised to show how the reasoner behaves in ambiguous situations and is due to Reiter [15].

| | |
|---|---|
| Quakers tend to be pacifist | $(Q(x) \succ\!\!- P(x))$ , |
| Republicans tend to be non-pacifist | $(R(x) \succ\!\!- \neg P(x))$ , |
| Nixon is a quaker | $(Q(nix))$ , |
| Nixon is a republican | $(R(nix))$ , |
| Is Nixon pacifist? | $(P(nix)?)$ . |

Clearly, there are three possible behaviors. The first, which is clearly undesirable, will give one of the two possibilities arbitrarily. The second, which is the

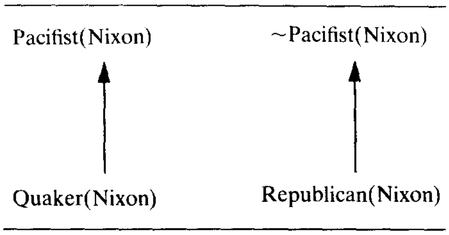


Fig. 2. Example 6.2.

behavior of reasoners using the inferential distance ordering instead of length of the path [20], will give two answers, leaving the decision to whomever uses the system. This kind of reasoner is called *credulous* because it gives good standing to all the possible conclusions. The last, the so-called *skeptical* reasoner, does not decide about ambiguity [5] by not giving any answer. Our reasoner is skeptical.

The context and defeasible rule set are

$$\mathcal{K} = \{R(nix), Q(nix)\}\,,$$

$$\Delta = \{Q(nix) >\!\!- P(nix),\ R(nix) >\!\!- \neg P(nix)\}$$

respectively (see Fig. 2). We have two argument structures, one for $P(nix)$ and one for $\neg P(nix)$, namely,

$$\langle T_1, P(nix)\rangle = \langle \{Q(nix) >\!\!- P(nix)\}, P(nix)\rangle\,,$$

$$\langle T_2, \neg P(nix)\rangle = \langle \{R(nix)\rangle >\!\!- \neg P(nix)\}, \neg P(nix)\rangle\,.$$

None of those argument structures defeats the other; they interfere and they are not ordered by specificity.

**Example 6.3** (*Cascaded Ambiguities*). This example is an extension of the Nixon Diamond constructed to show how simple-minded skeptical reasoners can be fooled to believe in the militarism (non-anti-militarism) of Nixon [5].

| | |
|---|---|
| Quakers tend to be pacifist | $(Q(x) >\!\!- P(x))$ , |
| Republicans tend to be non-pacifist | $(R(x) >\!\!- \neg P(x))$ , |
| Pacifists tend to be anti-military | $(P(x) >\!\!- A(x))$ , |
| Republicans tend to be football fans | $(R(x) >\!\!- F(x))$ , |
| Football fans tend to be non-anti-military | $(F(x) >\!\!- \neg A(x))$ , |
| Nixon is a quaker | $(Q(nix))$ , |
| Nixon is a republican | $(R(nix))$ , |
| Is Nixon anti-military? | $(A(nix)?)$ . |

The context and defeasible rule set are

$$\mathcal{K} = \{R(nix), Q(nix)\}\,,$$

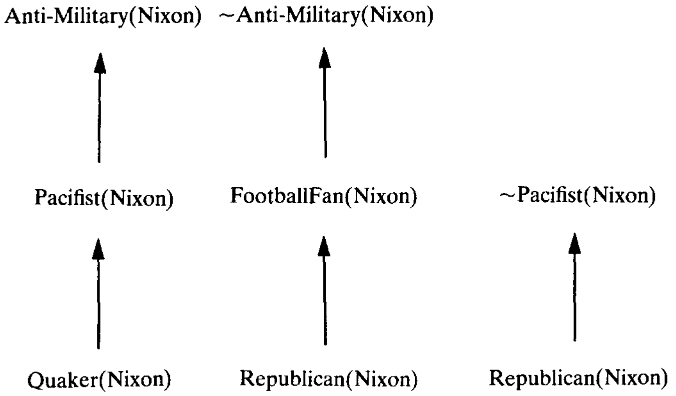
$$\begin{aligned}\Delta = \{&Q(nix) \mathrel{>\!\!-} P(nix), R(nix) \mathrel{>\!\!-} \neg P(nix), P(nix) \mathrel{>\!\!-} A(nix)\,,\\ &R(nix) \mathrel{>\!\!-} F(nix), F(nix) \mathrel{>\!\!-} \neg A(nix)\}\,.\end{aligned}$$

respectively (see Fig. 3). We have two argument structures, one for $A(nix)$ and one for $\neg A(nix)$, namely,

$$\begin{aligned}\langle T_1, A(nix)\rangle &= \langle\{Q(nix) \mathrel{>\!\!-} P(nix), P(nix) \mathrel{>\!\!-} A(nix)\}, A(nix)\rangle\,,\\ \langle T_2, \neg A(nix)\rangle &= \langle\{R(nix) \mathrel{>\!\!-} F(nix),\\ &\qquad F(nix) \mathrel{>\!\!-} \neg A(nix)\}, \neg A(nix)\rangle\,.\end{aligned}$$

Neither of those argument structures defeats the other and our reasoner remains skeptical.

Notice that some skeptical reasoners will consider the "path" (using inheritance reasoners terminology), $\{Q(nix) \mathrel{>\!\!-} P(nix), P(nix) \mathrel{>\!\!-} A(nix)\}$ as being preempted by $\{R(nix) \mathrel{>\!\!-} \neg P(nix)\}$ and hence leaving $\{R(nix) \mathrel{>\!\!-} F(nix), F(nix) \mathrel{>\!\!-} \neg A(nix)\}$ free to support the conclusion about Nixon being non-anti-military. That situation does not arise in our case. The argument structure $\langle\{R(nix) \mathrel{>\!\!-} \neg P(nix)\}, \neg P(nix)\rangle$ counterargues $\langle T_1, A(nix)\rangle$ at $P(nix)$, but $\langle\{R(nix) \mathrel{>\!\!-} \neg P(nix)\}, \neg P(nix)\rangle$ is not more specific than $\langle T_1, A(nix)\rangle$. Therefore, $\langle\{R(nix) \mathrel{>\!\!-} \neg P(nix)\}, \neg P(nix)\rangle$ does not defeat $\langle T_1, A(nix)\rangle$.



| Level | $T_1$ | $T_2$ | $T_3$ |
|---|---|---|---|
| 1 | IS | IS | IS |
| 2 | I | I | I |
| 3 | I | I | I |

Fig. 3. Example 6.3 (S = Supporting, I = Interfering).

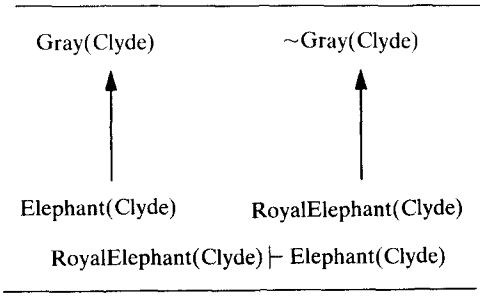


Fig. 4. Example 6.4.

**Example 6.4** (*Royal African Elephants*). This example deals with "on-path versus off-path preemption" and is due to Sandewall [16], in the context of inheritance reasoners.

| | |
|---|---|
| Elephants tend to be gray | $(E(x) \mathrel{>\!\!-} G(x))$ , |
| Royal elephants tend to be non-gray | $(R(x) \mathrel{>\!\!-} \neg G(x))$ , |
| Royal elephants are elephants | $(R(x) \supset E(x))$ , |
| African elephants are elephants | $(A(x) \supset E(x))$ , |
| Clyde is a Royal elephant | $(R(clyde))$ , |
| Clyde is an African elephant | $(A(clyde))$ , |
| Is Clyde non-gray? | $(\neg G(clyde))$ . |

The context and defeasible rules are

$$\mathcal{K} = \{R(clyde),\ A(clyde),\ R(clyde) \supset E(clyde),\ A(clyde) \supset E(clyde)\}\ ,$$

$$\Delta = \{E(clyde) \mathrel{>\!\!-} G(clyde),\ R(clyde) \mathrel{>\!\!-} \neg G(clyde)\}$$

respectively (see Fig. 4). We have three argument structures, two for $G(clyde)$ and one for $\neg G(clyde)$, namely,

$$\langle T_1, G(clyde)\rangle = \langle \{E(clyde) \mathrel{>\!\!-} G(clyde)\}, G(clyde)\rangle\ ,$$

$$\langle T_2, G(clyde)\rangle = \langle \{E(clyde) \mathrel{>\!\!-} G(clyde)\}, G(clyde)\rangle\ ,$$

$$\langle T_3, \neg G(clyde)\rangle = \langle \{R(clyde) \mathrel{>\!\!-} \neg G(clyde)\}, \neg G(clyde)\rangle\ .$$

Clearly, the more specific argument structure is $\langle T_3, \neg G(clyde)\rangle$, matching our intuitions.

**Example 6.5** (*Adult University Students*). This example deals with "defeasible specificity", which our system does not have, and is due to Geffner [2]. Geffner will draw a conclusion here, while we will not.

| | |
|---|---|
| Adults tend to work | $(A(x) \mathrel{>\!\!-} W(x))$ , |
| University students tend not to work | $(U(x) \mathrel{>\!\!-} \neg W(x))$ , |

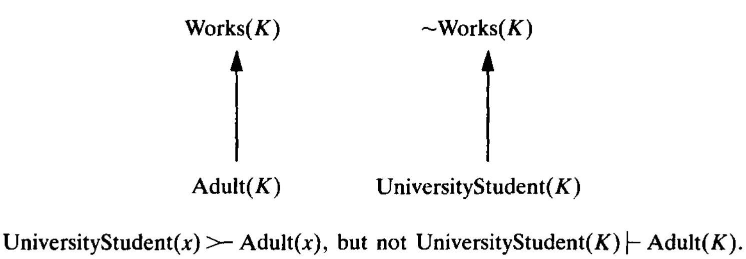

Works(K)
~Works(K)
Adult(K)
UniversityStudent(K)
UniversityStudent(x) >— Adult(x), but not UniversityStudent(K) |- Adult(K).

Fig. 5. Example 6.5.

| | |
|---|---|
| University students tend to be adults | $(U(x) \mathrel{>\!\!-} A(x))$ , |
| Ken is a university student | $(U(K))$ , |
| Ken is an adult | $(A(K))$ , |
| Does Ken not work? | $(\neg W(K))$ . |

The arguments are depicted in Fig. 5. Although $U(x) \mathrel{>\!\!-} A(x)$, this cannot be a part of either argument (first because it makes each argument nonminimal, and moreover, because it causes inconsistency in the second argument), so there is no specificity. Had the evidential context been only that Ken is a university student, from which $A(K)$ is derived, then the second argument would have been more specific.

**Example 6.6** (*Prima Facie Inconsistency of Rules*). This example deals with rules that are not "epsilon-sound" in the sense of Geffner and Pearl [2]. Their system cannot entertain such rules, while ours simply draws no conclusion.

$$(P \mathrel{>\!\!-} Q)\,, \quad (P \mathrel{>\!\!-} \neg R)\,, \quad (Q \mathrel{>\!\!-} R)\,,$$
$$(P)\,, \quad (R \text{ or } \neg R)?\,.$$

The arguments are depicted in Fig. 6. $P$ activates the first argument, as does

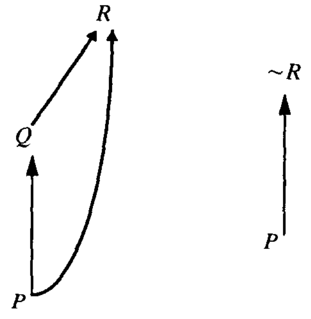


Fig. 6. Example 6.6. $P$ activates $T_1$, as does $P \wedge Q$. Each suffices for $T_2$. Meanwhile $P$ activates $T_2$, which suffices for $T_1$. Neither is more specific.

$P \wedge Q$. Each suffices for the second argument. Meanwhile, $P$ activates the second argument, which suffices for activating the first argument. Neither is more specific; no conclusion is justified.

**Example 6.7** (*Yale Shootings*). This example deals with the Yale Shooting example and its extension by Hanks and McDermott [4] (see Fig. 7). They note that Poole's original system can choose the second argument over the first argument, but complain that it cannot block the first argument once extended. Blocking the extended argument, as desired, is trivial in this system. This was also true of the system in [7]. Moreover, the present system permits the second argument to defeat the first even when the final rule is replaced by a material conditional.

| | |
|---|---|
| Aliveness tends to persist | $(\mathit{Alive}@t \mathrel{>\!\!-} \mathit{Alive}@t+1)$ , |
| Loadedness tends to persist | $(\mathit{Loaded}@t \mathrel{>\!\!-} \neg \mathit{Loaded}@t+1)$ , |
| Firing a loaded gun coerces a change in Aliveness | $(\mathit{Alive}@t \wedge \mathit{Fired}@t \wedge \mathit{Loaded}@t \mathrel{>\!\!-} \neg \mathit{Alive}@t+1)$ , |
| Fred is Alive | $(\mathit{Alive}@0)$ , |
| The gun is loaded | $(\mathit{Loaded}@1)$ , |
| The gun is fired | $(\mathit{Fired}@2)$ , |
| Does Fred die? | $(\neg \mathit{Alive}@3)$ . |

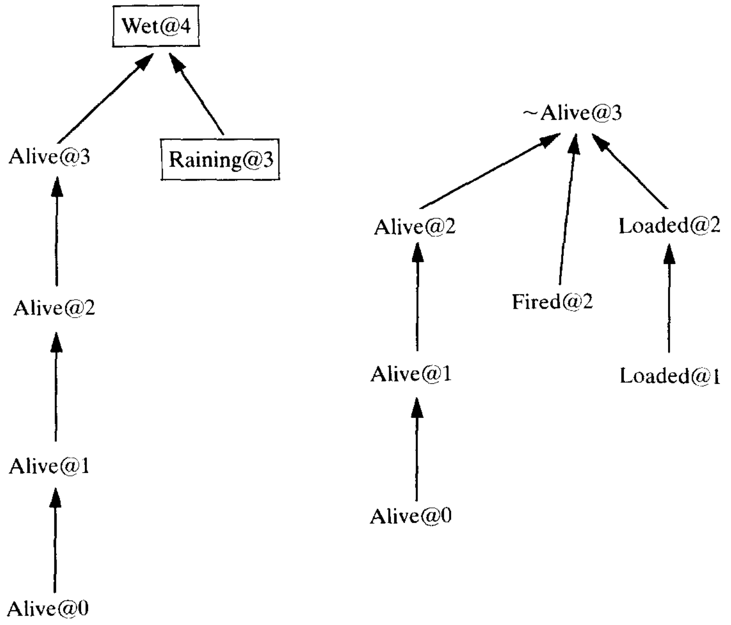


Fig. 7. $T_2$ defeats $T_1$, and it will defeat the proposed extension of $T_1$. The top rule in $T_2$, Loaded@2 $\wedge$ Fired@2 $\wedge$ Alive@2 $\mathrel{>\!\!-}$ ~Alive@3 could be *replaced* by a material conditional, Loaded@2 $\wedge$ Fired@2 $\wedge$ Alive@2$\rightarrow$~Alive@3 and there would still be defeat.

Clearly the theory

$$\begin{aligned}\langle T_1, \mathit{Alive@3}\rangle = \langle\{&\mathit{Alive@0} \succ\!\!- \mathit{Alive@1},\ \mathit{Alive@1} \succ\!\!- \mathit{Alive@2},\\ &\mathit{Alive@2} \succ\!\!- \mathit{Alive@3}\},\ \mathit{Alive@3}\rangle\end{aligned}$$

is defeated by the theory

$$\begin{aligned}\langle T_2, \neg\mathit{Alive@3}\rangle = \langle\{&\mathit{Alive@0} \succ\!\!- \mathit{Alive@1},\ \mathit{Alive@1} \succ\!\!- \mathit{Alive@2},\\ &\mathit{Alive@2} \succ\!\!- \mathit{Alive@3},\\ &\mathit{Loaded@1} \succ\!\!- \mathit{Loaded@2},\ \mathit{Loaded@2} \succ\!\!- \mathit{Loaded@3},\\ &\mathit{Alive@2} \wedge \mathit{Fired@2} \wedge \mathit{Loaded@2} \succ\!\!- \neg\mathit{Alive@3}\},\\ &\neg\mathit{Alive@3}\rangle\,.\end{aligned}$$

In fact, if the material conditional

$$\mathit{Alive@t} \wedge \mathit{Fired@t} \wedge \mathit{Loaded@t} \supset \neg\mathit{Alive@t} + 1$$

is included in the necessary part of the evidence, then the theory

$$\begin{aligned}\langle T_{2b}, \neg\mathit{Alive@3}\rangle = \langle\{&\mathit{Alive@0} \succ\!\!- \mathit{Alive@1},\ \mathit{Alive@1} \succ\!\!- \mathit{Alive@2},\\ &\mathit{Alive@2} \succ\!\!- \mathit{Alive@3},\\ &\mathit{Loaded@1} \succ\!\!- \mathit{Loaded@2},\ \mathit{Loaded@2} \succ\!\!- \mathit{Loaded@3}\},\\ &\neg\mathit{Alive@3}\rangle\end{aligned}$$

defeats the first theory. Extending the first theory with the rule

$$\mathit{Alive@3} \wedge \mathit{Raining@3} \succ\!\!- \mathit{Wet@4}\,,$$

and the evidence *Raining@*3 just produces a theory that is defeated by the counterargument $T_2$ (or $T_{2b}$).

## 7. A justification finder

Implementations of defeasible reasoners are rarely seen beasts. An early attempt to introduce defeasible reasoning programming with specificity was Nute's d-Prolog [10, 11]. The language of d-Prolog provides facilities to define *absolute rules*, "every bat is a mammal", *defeasible. rules*, "birds fly", and *defeater rules*, "sick birds do not fly". The purpose of defeater rules was to account for the exceptions to defeasible rules. For instance, given the defeasible rule "birds fly", the defeater rule "sick birds do not fly" will stop us from concluding that "Tweety flies", in the presence of the fact "Tweety is a sick bird".

### *7.1. The language implemented*

The basic ideas of logic programming are introduced here using the standard notation for them. We will slightly modify that notation as we introduce our language.

**Definition 7.1.** *A definite clause* is a clause of the form:

$$B \Longleftarrow A_1, \ldots, A_n$$

with only one atom as the consequent. The consequent $B$ is called the *head* and the antecedent $A_1, \ldots, A_n$ is called the *body* of the definite clause.

It is customary to regard all clauses as implications, even though they have no head or body. We will alter this for our language in a way that is consistent with this presentation. The reasons for that modification will be given below.

**Definition 7.2.** A *definite goal* is a clause of the form:

$$\Longleftarrow A_1, \ldots, A_n ,$$

i.e., a definite clause with an empty consequent. The $A_i$ are sometimes called *subgoals* of the goal.

A *unit clause* is a clause of the form:

$$B \Longleftarrow ,$$

i.e., a definite clause with an empty body. We will alter this representation introducing the special atom *true*. Our unit clauses will be written:

$$B \Longleftarrow \mathit{true} .$$

Unit clauses are also called *facts*.

**Definition 7.3.** A *Horn clause* is a clause which is either a definite clause or a definite goal.

We have extended the representation in two ways. First, we added defeasible clauses, and second, we introduced a relation *neg* used to represent negative facts.

**Definition 7.4.** A *defeasible clause* is a clause of the form:

$$B \mathrel{-\!\!<} A_1, \ldots, A_n$$

with only one atom as the consequent. The consequent $B$ is called the *head* of the defeasible rule and the antecedent $A_1, \ldots, A_n$ is called the *body* of the defeasible clause.

The *neg* relation will allow the representation of negative facts in the system. Negation is handled in the same way as proposed by Nute [10, 11]. This relation is not related in any way to *negation as failure* and its only meaning is to refer to a negative fact. Negative facts relate to positive facts in the usual way. The system will treat the relation *neg* as a prefix forming part of the "name" of the atom and not as an operator. The system will recognize that *neg neg A* = *A*. That is, the goal *neg A* will be assumed as a consequence of a set $R$ of definite and defeasible clauses if and only if *neg A* is deducible from $R$ via a finite number of applications of modus ponens. The goal *neg neg A* will be assumed as a consequence of a set $R$ of definite and defeasible clauses if and only if *neg neg A*, or $A$, is deducible from $R$ via a finite number of applications of modus ponens. Thus, the relation *neg* does not have any special status; the system will treat the atom *neg A* in the same way as any other atom $C$.

Our *neg* operator can appear in the head of the rules, defeasible and otherwise. For instance,

$$\begin{aligned} &\mathit{neg}\ A \Longleftarrow \mathit{true}\,, \\ &\mathit{neg}\ A \Longleftarrow \mathit{neg}\ B, C\,, \\ &\mathit{neg}\ A \prec B, C, \mathit{neg}\ D\,, \end{aligned}$$

are legal rules. Notice that the first rule is asserting a negative fact.

**Definition 7.5.** A *knowledge base* $\mathbb{K}$ is a finite set of definite clauses and defeasible clauses, possibly containing atoms affected by the *neg* relation. A knowledge base is the equivalent of what previously was called a defeasible logic structure. In a knowledge base $\mathbb{K}$ the set $\mathscr{K}$ will be represented using definite clauses, and the set $\Delta$ will be represented using defeasible clauses.

### *7.2. Finding justifications*

The interpreter will work following the lines of the proof of Theorem 4.16 taking advantage of the inference mechanism of Prolog.

The input to **jf** is a knowledge base $\mathbb{K}$, and a ground query $Q$. The contents of the knowledge base were described in Section 7.1. A ground query $Q$ is a ground instance of an atom, possibly affected with the prefix *neg*. The justifier is invoked by issuing the command:

$$\mathit{analyze}(Q)\,,$$

which will start the process of testing whether there is an undefeated argument which supports $Q$ from the contents of $\mathbb{K}$.

If the search finds a justification the output of the system for such a query will be one of the argument structures that are justifying $Q$, and all the possible defeaters that were considered. All the justifiers can be obtained by rejecting the answer, and forcing the system to keep searching.

If the answer is negative, the system will have two possible answers. The query $Q$ has no supporting argument. Or even though arguments can be constructed to support it, all of them were defeated. In the latter case, the system will return all the potential justifiers, already defeated, with its associated defeaters. We will disregard the uninteresting case when $Q$ has no supporting argument.

The process begins by attempting to construct an argument for the given query $Q$. Arguments for $Q$ are constructed by using backward chaining over the knowledge base. We will follow Shapiro's [19] terminology. A *ground reduction* of a goal $G$ in a knowledge base $\mathbb{K}$ is the replacement of $G$ by the body of a ground instance of a clause (definite or defeasible), whose head is identical to $G$. A *defeasible inference tree* consists of nodes and edges which represent the goals reduced during the construction. The root of the tree is the original query and the nodes are the goals reduced during the backward chaining. Edges represent the relation between the head of the rule used in the reduction and the atoms in the body of that rule. The backward chaining on a node $G$ stops whenever $G$ is supported by a unit clause, i.e., a clause like $G \Longleftarrow \mathit{true}$. The following example will help to describe the process:

**Example 7.6.** Assume the following knowledge base $\mathbb{K}$:

| | |
|---|---|
| $\mathit{flies}(x) \mathrel{-\!\!<} \mathit{bird}(x)$ | (usually, birds fly) |
| $\mathit{neg\ flies}(x) \mathrel{-\!\!<} \mathit{penguin}(x)$ | usually, penguins do not fly) |
| $\mathit{bird}(x) \Longleftarrow \mathit{penguin}(x)$ | (penguins are birds) |
| $\mathit{penguin}(\mathit{opus}) \Longleftarrow \mathit{true}$ | (opus is a penguin) |

After the query "*analyze*(*flies*(*opus*))", the system will form the argument

$$\{\mathit{flies}(\mathit{opus}) \mathrel{-\!\!<} \mathit{bird}(\mathit{opus}),$$
$$\mathit{bird}(\mathit{opus}) \Longleftarrow \mathit{penguin}(\mathit{opus}),\ \mathit{penguin}(\mathit{opus}) \Longleftarrow \mathit{true}\}$$

by backward chaining from *flies*(*opus*).

The system will always form the most specific argument. If the system is forced to backtrack from a unit clause $G \Longleftarrow \mathit{true}$, it will not attempt to find support for $G$ in other clauses. Following those clauses will only produce a less specific argument. This observation was already suggested in the proof of Theorem 4.16.

After forming an argument, the system will try to find counterarguments for the recently formed argument by backward chaining from the negation of atoms in the original argument. Actually, the system will form a set with the atoms in the argument, and will add to that set any atom that is derivable from those atoms and the definite clauses in $\mathbb{K}$. For instance, in the example above, it will find the counterargument

$$\{\mathit{neg\ flies}(\mathit{opus}) \mathrel{-\!\!<} \mathit{penguin}(\mathit{opus}),\ \mathit{penguin}(\mathit{opus}) \Longleftarrow \mathit{true}\}\ .$$

Finally, the system will test the argument and the counterargument for specificity using models of argument activation (see Simari [18]) and models of the counterargument. In short, $M$ is an activation model for $\langle T, h\rangle$ if $M$ is a model of $\mathcal{K}_N$ and $M$ is also a model for some $e \in Sent_C(\mathcal{L})$ and for the rules that form the subset $T'$ of $T$ such that $\mathcal{K}_N \cup \{e\} \cup T' \mathrel{|\!\!\sim} h$. Using those criteria in our example, we will find that any activation model for the argument for *neg flies*(*opus*) is an activation model for the argument for *flies*(*opus*). But there is an activation model for *flies*(*opus*) which is not an activation model for *neg flies*(*opus*), namely the one where *bird*(*opus*) is true but *penguin*(*opus*) is not.

If the argument is defeated, as is the case in the example, the system will backtrack in the process that formed the original argument, discarding the last rule added to the tree, trying to replace it with another. If it finds one, the process of finding and testing defeaters is repeated. Otherwise, further backtracking is necessary. This process will continue until an undefeated argument is produced or all the backtracking possibilities are exhausted.

## 8. Conclusions

In this paper we have presented a mathematical approach to defeasible reasoning. This approach is based on the notion of specificity introduced by Poole and the general theory of warrant as presented by Pollock. Poole's approach to specificity was correct but he stopped short of presenting a complete approach to it. We proved that an order relation can be introduced among equivalence classes under the equi-specificity relation. Poole did not pursue operational aspects of applying specificity. We did that here.

Pollock has suggested an operational framework for performing reasoning, but he dismissed useful and prevalent generalizations of specificity. Taking his definition of warrant, we have applied it and transformed it into a justification schema which defines the set of justified facts from a given defeasible logic structure. One result of this paper is a theorem that ensures the termination of the process of finding the justified facts. The proof of that result is based on the order relation mentioned above.

In order to implement the theoretical ideas, a suitable restriction of the language has been defined. The language used to represent the context $\mathcal{K}$ has been restricted to a subset of first-order logic, Horn clauses, and the language used to represent defeasible rules in $\Delta$ has been restricted in a similar way, to a Horn-clause-like syntax. The interpreter was written in Prolog, and running on top of it provides a defeasible reasoning tool for Prolog.

The implementation of the system has taken advantage of the theoretical findings. The general mechanism used in the implementation to find justifications is based on the structures built in proof of the theorem on termination. The process used to compare two argument structures for specificity is based on semantical work that is not reported here.[6] Two more lemmas (Lemmas 2.24 and 2.25) define a reduced search space. Meanwhile, it is the prospect of implementation that suggested many of these theorems.

In comparison with inheritance, this system generalizes the idea of path, clarifies the logic of reinstatement, and even in its Horn clause form, provides more expressive language. In comparison with [7], this system shares the same spirit and many of its syntactic considerations, though reproduces almost none of the details. In particular, $\mathcal{K}$, $\Delta$, and $\succ$ are taken from [7], which in turn originates with Kyburg [6]. Further, [7]'s definition of arguments as digraphs confuses definitional and implementational issues, which this paper separates. In comparison with Geffner, this approach represents an alternative, older paradigm, based on arguments instead of irrelevance.

To summarize, the introduction of defeasible logic structures as a way of performing defeasible reasoning represents the unification of ideas in a formal and concise system which exhibits a correct, and uniform, behavior when applied to the benchmark examples in the literature. The investigation of the theoretical issues has aided the study of how this kind of reasoner can be realized on a computer, leading to an efficient implementation. We believe that the presentation here may have more permanence than past approaches to defeasible argument.

## Acknowledgement

This research has been conducted within the Washington University Center for Intelligent Computer Systems, an industrial consortium supported by McDonnell Douglas Corporation and Southwestern Bell Corporation. Final preparation supported by National Science Foundation R-9008012. Karl Stiefvater found many flaws, some of which have been fixed.

## References

[1] J.P. Delgrande, An approach to default reasoning based on a first-order conditional logic, in: *Proceedings AAAI-81*, Seattle, WA (1987).

[2] H.A. Geffner and J. Pearl, A framework for reasoning with defaults, Tech. Report TR-94III CSD-870058, Cognitive Systems Lab., University of California, Los Angeles, CA (1989); also in: H.E. Kyburg, R.P. Loui and G. Carlson, eds., *Knowledge Representation and Defeasible Reasoning* (Kluwer Academic Publishers, London, 1990) 245–265.

[3] C. Glymour and R.H. Thomason, Default reasoning and the logic of theory perturbation, in: *Proceedings Nonmonotonic Reasoning Workshop*, Menlo Park, CA (1984) 93–102.

[4] S. Hanks and D. McDermott, Nonmonotonic logic and temporal projection, *Artif. Intell.* **33** (3) (1987) 379–412.

[6] Available in the dissertation [17].

[5] J.F. Horty, R.H. Thomason and D.S. Touretzky, A skeptical theory of inheritance in nonmonotonic semantic networks, Tech. Report CMU-CS-87-175, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA (1987); also: *Artif. Intell.* **42** (1990) 311–348.

[6] H.E. Kyburg, *Logical Foundations of Statistical Inference* (Reidel, Dordrecht, Netherlands, 1974).

[7] R.P. Loui, Defeat among arguments: a system of defeasible inference, *Comput. Intell.* **3** (3) (1987) 100–106.

[8] E. Neufeld, D. Poole and R. Aleliunas, Probabilistic semantics and defaults, in: R.D. Schachter, T.S. Levitt, L.N. Kanal and J.F. Lemmer, eds., *Uncertainty in AI* **4** (North-Holland, Amsterdam, 1990) 121–131.

[9] D. Nute, A non-monotonic logic based on conditional logic, Tech. Report ACMC 01-0007, University of Georgia, Athens, GA (1985).

[10] D. Nute, Defeasible reasoning, in: J.H. Fetzer, ed., *Aspects of Artificial Intelligence* (Kluwer Academic Publishers, Norwell, MA, 1988) 251–288.

[11] D. Nute and M. Lewis, d-Prolog: a users manual, Tech. Report ACMC 01-0017, University of Georgia, Athens, GA (1986).

[12] J.L. Pollock, Defeasible reasoning, *Cogn. Sci.* **11** (1987) 481–518.

[13] D. Poole, A logical framework for default reasoning, *Artif. Intell.* **36** (1) (1988) 27–47.

[14] D. Poole, On the comparison of theories: preferring the most specific explanation, in: *Proceedings IJCAI-85*, Los Angeles, CA (1985) 144–147.

[15] R. Reiter, A logic for default reasoning, *Artif. Intell.* **13** (1980) 81–132.

[16] E. Sandewall, Non-monotonic inference rules for multiple inheritance with exceptions, *Proceedings IEEE* **74** (1986) 481–518.

[17] G.R. Simari, A mathematical treatment of defeasible reasoning and its implementation, Ph.D. Thesis, Department of Computer Science, Washington University, St. Louis, MO (1989).

[18] G.R. Simari, On the logic of defeasible reasoning, Tech. Report WUCS-89-12, Department of Computer Science, Washington University, St. Louis, MO (1989).

[19] L. Sterling and E. Shapiro, *The Art of Prolog* (MIT Press, Cambridge, MA, 1986).

[20] D.S. Touretzky, *The Mathematics of Inheritance Systems* (Morgan Kaufmann, Los Altos, CA, 1986).