Dear Author

Here are the proofs of your article.

- You can submit your corrections **online,** via **e-mail** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- You can also insert your corrections in the proof PDF and **email** the annotated PDF.
- For **fax** submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the **journal title**, **article number**, and **your name** when sending your response via e-mail or fax.
- **Check** the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- **Check** the questions that may have arisen during copy editing and insert your answers/corrections.
- **Check** that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript*.
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style.
- Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections **within 48 hours**, we will send you a reminder.
- Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI. **Further changes are, therefore, not possible.**
- The **printed version** will follow in a forthcoming issue.

Please note

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL:

`http://dx.doi.org/10.1007/s10472-015-9471-9`

If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information, go to: http://www.link.springer.com.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us, if you would like to have these documents returned.

AUTHOR'S PROOF

Metadata of the article that will be visualized in OnlineFirst

1	Article Title	**A series of revisions of David Poole's specificity**	
2	Article Sub-Title		
3	Article Copyright - Year	**Springer International Publishing Switzerland 2015 (This will be the copyright line in the final PDF)**	
4	Journal Name	Annals of Mathematics and Artificial Intelligence	
5	Corresponding Author	Family Name	**Wirth**
6		Particle	
7		Given Name	**Claus-Peter**
8		Suffix	
9		Organization	Harz University of Applied Sciences
10		Division	Department of Automation and Computer Sciences
11		Address	Friedrichstr. 57–59, 38855, Wernigerode, Germany
12		e-mail	wirth@logic.at
13	Author	Family Name	**Stolzenburg**
14		Particle	
15		Given Name	**Frieder**
16		Suffix	
17		Organization	Harz University of Applied Sciences
18		Division	Department of Automation and Computer Sciences
19		Address	Friedrichstr. 57–59, 38855, Wernigerode, Germany
20		e-mail	fstolzenburg@hs-harz.de
21	Schedule	Received	
22		Revised	
23		Accepted	
24	Abstract	In the middle of the 1980s, David Poole introduced a semantic, model-theoretic notion of specificity to the artificial-intelligence community. Since then it has found further applications in non-monotonic reasoning, in particular in defeasible reasoning. Poole tried to approximate the intuitive human concept of specificity, which seems to be essential for reasoning in everyday life with its partial and inconsistent information. His notion, however, turns out to be intricate and problematic, which — as we show — can be overcome to some extent by a closer approximation of the intuitive human concept of specificity. Besides the intuitive advantages of our novel specificity orderings over Poole's specificity relation in the classical examples of the literature, we also report some hard mathematical facts: Contrary to what was claimed before, we show that Poole's relation is not transitive in general. The first of our specificity orderings (CP1) captures Poole's original intuition as close as we could get after the correction of its technical flaws. The second one (CP2) is a variation of CP1 and	

Please note: Images will appear in color online but will be printed in black and white.

presents a step toward similar notions that may eventually solve the intractability problem of Poole-style specificity relations. The present means toward deciding our novel specificity relations, however, show only slight improvements over the known ones for Poole’s relation; therefore, we suggest a more efficient workaround for applications in practice.

25	Keywords separated by ' - '	Artificial intelligence - Non-monotonic reasoning - Defeasible reasoning - Specificity - Positive-conditional specification
26	Foot note information	

Ann Math Artif Intell
DOI 10.1007/s10472-015-9471-9

A series of revisions of David Poole's specificity

Claus-Peter Wirth[1] · Frieder Stolzenburg[1]

Abstract In the middle of the 1980s, David Poole introduced a semantic, model-theoretic notion of specificity to the artificial-intelligence community. Since then it has found further applications in non-monotonic reasoning, in particular in defeasible reasoning. Poole tried to approximate the intuitive human concept of specificity, which seems to be essential for reasoning in everyday life with its partial and inconsistent information. His notion, however, turns out to be intricate and problematic, which — as we show — can be overcome to some extent by a closer approximation of the intuitive human concept of specificity. Besides the intuitive advantages of our novel specificity orderings over Poole's specificity relation in the classical examples of the literature, we also report some hard mathematical facts: Contrary to what was claimed before, we show that Poole's relation is not transitive in general. The first of our specificity orderings (CP1) captures Poole's original intuition as close as we could get after the correction of its technical flaws. The second one (CP2) is a variation of CP1 and presents a step toward similar notions that may eventually solve the intractability problem of Poole-style specificity relations. The present means toward deciding our novel specificity relations, however, show only slight improvements over the known ones for Poole's relation; therefore, we suggest a more efficient workaround for applications in practice.

Keywords Artificial intelligence · Non-monotonic reasoning · Defeasible reasoning · Specificity · Positive-conditional specification

Q3 **Mathematics Subject Classification (2010)**

✉ Claus-Peter Wirth
Q1 wirth@logic.at

Frieder Stolzenburg
fstolzenburg@hs-harz.de

Q2 [1] Department of Automation and Computer Sciences, Harz University of Applied Sciences, Friedrichstr. 57–59, 38855 Wernigerode, Germany

1 Introduction

A possible explanation of how humans manage to interact with reality — in spite of the fact that their information on the world is partial and inconsistent — mainly consists of the following two points:

1. Humans use a certain amount of *rules for default reasoning* and are aware that some arguments relying on these rules may be *defeasible*.
2. In case of the frequent conflicting or even contradictory results of their reasoning, they *prefer more specific arguments* to less specific ones.

An intuitive concept of specificity plays an essential rôle in this explanation, which is interesting because it seems to be highly successful in practice, even if it were just an epiphenomenon providing an *ex eventu* explanation of human behavior.

On the long way approaching the proven intuitive human concept of specificity, the first milestone marks the development of a semantic, model-theoretic notion of specificity having passed first tests of its usefulness and empirical validity. Indeed, at least as the first step, a semantic, model-theoretic notion will probably offer a broader and better basis for applications in systems for common-sense reasoning than notions that depend on peculiarities of special calculi or even on extra-logical procedures. This holds in particular if the results of these systems are to be accepted by humans.

David Poole has sketched such a notion as a binary relation on arguments and evaluated its intuitive validity with some examples in [22]. Poole's notion of specificity was given a more appropriate formalization in [26]. The properties of this formalization were examined in detail in [27].

In Sections 2 and 3, we recall basic notions and notation and the elementary motivating examples.

In Section 4, we present a detailed analysis of the reasons behind our intuition that Poole's specificity is a first step on the right way.

We expect that the results of this detailed analysis will carry us even beyond this paper to future improved concepts of specificity, especially w.r.t. efficiency, but also w.r.t. intuitive adequacy. We hope that the closer we get to human intuition, the more efficiently our concepts can be implemented, simply because they seem to run so well on the human hardware, which — by all that we know today — is pretty slow.

In Section 5, we specify formal requirements on any reasonably conceivable relation of specificity.

In Section 6, we disambiguate Poole's specificity relation from slightly improved versions, such as the one in [26], and introduce a *novel specificity ordering (CP1)*, a *correction* of Poole's specificity in the sense that it removes a crucial shortcoming of Poole's original relation (P1) and its slight improvements (P2, P3), namely their *lack of transitivity.*

In Section 7, we present several *examples* that are to convince the carefully contemplating reader of the superiority of our novel specificity relation CP1 w.r.t. human intuition.

In Section 8, we discuss *efficiency issues*. We introduce a further *novel specificity ordering (CP2)* (a variation of CP1) as a first step toward similar notions that may finally solve the intractability problem of Poole-style specificity relations. The present means toward deciding our novel specificity relations, however, show only slight improvements over the known ones for Poole's relation; therefore, we suggest a more efficient workaround for applications in practice.

In Section 9, we draw some first conclusions.

2 Basic notions and notation

Definition 1 (Term, Atom)
A *term* is inductively defined to be either a function symbol applied to a (possibly empty) list of terms or a symbol for a free variable.

Q4 An *atom* consists of a predicate symbol applied to a (possibly empty) list of terms.

In what follows, we will mainly use nullary function symbols ("constants"), such as tweety, and singulary predicate symbols, such as bird, forming atoms such as bird(tweety), which states that tweety is a bird.

2.1 Specifying rules and their theories

For the remainder of this paper, let us narrow the general logical setting of specificity down to the concrete framework of *defeasible logic with the restrictions of positive-conditional specification with an inactive negation symbol*, as found e.g. in [27] and [5].

In effect, these restrictions give us the standard "definite rules" of positive-conditional specification (or Horn-clause logic). Positive-conditional specification differs from logic programming in PROLOG (cf. e.g. [6, 18]) insofar as termination issues and the order of the definite clauses are irrelevant for the semantics, and insofar as there is no cut predicate ('!') and no negation as failure.

Such *definite rules* are implications of the following form: The conclusion is an atom; the condition is a (possibly empty) conjunction of (positive) atoms which may contain extra variables (i.e. free variables not occurring in the conclusion). This is can be seen as quantifier-free first-order logic with specifications restricted to implications of the mentioned form.

We ask the reader not to get confused on the mentioned effective form of our rules by the fact that — in place of the atoms — literals resulting from an inactive negation symbol are actually admitted in the rules of Definition 2 (see below). This special form of negation is standard in defeasible logic for convenience in the application context (such as an argumentation framework). In this paper, however, we can consider this negation just as a form of syntactic sugar (cf. Definition 3, Remark 1).

Definition 2 (Literal, Rule)
A *literal* is an atom, possibly prefixed with the symbol "$\neg$" for negation.

A *rule* is a literal, but possibly suffixed with a reverse implication symbol "$\Leftarrow$" that is followed by a conjunction of literals, consisting of one literal at least.

Definition 3 (Theory, Derivation)
Let Π be a set of rules. The *theory of* Π is the set $\mathfrak{T}_{\Pi}$ inductively defined to contain

– all instances of literals from Π and

– all literals L for which there is a conjunction C of literals from $\mathfrak{T}_{\Pi}$ such that $L \Leftarrow C$ is an instance of a rule in Π.

For $\mathfrak{L} \subseteq \mathfrak{T}_{\Pi}$, we also say that Π *derives* $\mathfrak{L}$, and write $\Pi \vdash \mathfrak{L}$.

2.2 Secondary aspects of our logic

Remark 1 (Negation Symbol "¬")
The negation symbol "¬, which occurs in Definition 2 and which seemingly gets us beyond the definite rules of positive-conditional specification s by admitting literals instead of just atoms, does not have any effect on the *derivations* and *theories* considered in this paper (cf. Definition 3). For instance, the literal ¬flies(edna) may actually be considered as the atom resulting from application of the predicate ¬flies to the constant symbol edna.

On the other hand, if we write an atom A as $A = \mathsf{true}$, and a negated atom $\neg A$ as the equational atom $A = \mathsf{false}$, for the data type Boolean given by the constructors true and false, then the rules of our specification can be seen as *positive*-conditional equational specifications in the framework for *positive/negative*-conditional equational specification found in [33], and [28, 29].

In the application context, of course, the literals ¬flies(edna) and flies(edna) will be considered to be *contradictory* (cf. Definition 4), but this is a secondary and non-essential notion built on top of our derivations and theories, which do not rely on this notion.

As a consequence, none of the results in this paper relies on this special negation symbol. To the contrary, in the weakness of our logical theories we see an indication for the generality of our results (cf. Remark 2).

To distinguish the inactive negation here from negation as failure and from any other form of negation playing an active rôle in derivation, the symbol "∼" is sometimes used in the literature of defeasible logic in place of our more standard symbol "¬".

Definition 4 (Contradictory Sets of Rules)
A set of rules Π is called *contradictory* if there is an atom A such that $\Pi \vdash \{A, \neg A\}$; otherwise Π is *non-contradictory*.

Remark 2 (Weakness of Our Logical Theories)
On the one hand, $\{\mathsf{A}, \neg\mathsf{A} \Leftarrow \mathsf{A}\}$ is contradictory according to Definitions 3 and 4. On the other hand, $\{\mathsf{A} \Leftarrow \neg\mathsf{A}, \neg\mathsf{A} \Leftarrow \mathsf{A}\}$ is non-contradictory according to these definitions, although we can infer both A and ¬A from $\{\mathsf{A} \Leftarrow \neg\mathsf{A}, \neg\mathsf{A} \Leftarrow \mathsf{A}\}$ in classical (i.e. two-valued) logic. For the case of our very limited formal language, our notions of consequence and contradiction are equivalent both to intuitionistic logic and to the three-valued logic where ¬ and ∧ are given as usual, but (following neither Kleene nor Łukasiewicz) implication has to be defined via $(A \Leftarrow \mathsf{TRUE}) = A$, $(A \Leftarrow \mathsf{FALSE}) = \mathsf{TRUE}$, $(A \Leftarrow \mathsf{UNDEF}) = \mathsf{TRUE}$.

2.3 Global parameters for the given specification

Throughout this paper, we will assume a set of literals Π^{F} and two sets of rules Π^{G}, Δ (cf. Definition 2) to be given:

– A set Π^{F} of literals meant to describe the *facts* of the concrete situation under consideration,

- a set Π^{G} of *general rules* meant to hold in all possible worlds,[1] and
- a set Δ of *defeasible* (or default) rules meant to hold in most situations.

The set $\Pi := \Pi^{\mathrm{F}} \cup \Pi^{\mathrm{G}}$ is the set of *strict* rules that — contrary to the defeasible rules — are considered to be safe and are not doubted in the concrete situation.

2.4 Formalization of arguments

Whether a rule is a strict one from Π or a defeasible one from Δ has no effect on theories and derivations (cf. Definition 3). If a contradiction occurs, however, we will narrow the defeasible rules from Δ down to a subset $\mathcal{A}$ of its *ground* instances (i.e. instances without free variables) — such that no further instantiation can occur. Such a subset, together with the literal whose derivation is in focus, is called an *argument*. With implicit reference to the given sets of rules Π and Δ, the formal definition is as simple as follows.

Definition 5 ([Contradictory] [Minimal] Argument)
$(\mathscr{A}, L)$ is an *argument* if $\mathscr{A}$ is a set of ground instances of rules from Δ and $\mathscr{A} \cup \Pi \vdash \{L\}$.
$(\mathscr{A}, L)$ is a *minimal argument* if $\mathscr{A}$ is an argument, but $(\mathscr{A}', L)$ is not an argument for any proper subset $\mathscr{A}' \subsetneq \mathscr{A}$.
An argument $(\mathscr{A}, L)$ is *contradictory* if $\mathscr{A} \cup \Pi$ is a contradictory set of rules.

Remark 3 (Non-Ground Arguments)
From a refined standpoint, what we actually need is not exactly a set $\mathscr{A}$ of *ground* instances, but just of the instances applied in the derivation. Then, however, we have to freeze the variables in $\mathscr{A}$ because they must not be instantiated in the derivation $\mathscr{A} \cup \Pi \vdash \{L\}$. We avoid this refinement here until we come to Section 8.3, because it does not play an essential rôle before and because we want to stay within the traditional framework as long as possible to facilitate a more direct comparison.

Remark 4 (Minimality and Non-Contradiction of Arguments)
Some authors (cf. e.g. [5, 27]) require all arguments

1. to be minimal arguments, and
2. to be non-contradictory.

Because non-minimal as well as contradictory arguments often occur in practical situations, there is no use-oriented justification for any of these requirements.

For requirement 1 there is no conceptual justification, either, because the non-minimal arguments become inessential by our preference on specific arguments, in the sense that for every argument there must be a minimal sub-argument that is at least as specific, cf. Corollaries 3, 5, and 8. Because being contradictory is only a secondary aspect of our logic (cf. Section 2.2), there is no conceptual justification for requirement 2, either.

To obtain a more general setting in the comparison of arguments, we omit these restrictions in the context of this paper, where they turned out to be completely superfluous. In particular, the omission of these requirements has no effect on the results of this paper.

[1] In the approach of [27], the set Π^{G} must not contain mere literals (without suffixed condition), also called *presumptions*. To obtain a more general setting, we omit this additional restriction in the context of this paper, simply because it is neither intuitive nor required for our framework here. For the actual occurrence of a literal in Π^{G}, see the discussion of Example 18 in Section 7.4.

2.5 Quasi-Orderings

We will use several binary relations comparing arguments according to their specificity. For any relation written as $\lesssim_N$ ("being more or equivalently specific w.r.t. N"), we set

$$\begin{aligned}
\gtrsim_N &:= \{(X,Y) \mid Y \lesssim_N X\} && \text{("less or equivalently specific"),}\\
\approx_N &:= \lesssim_N \cap \gtrsim_N && \text{("equivalently specific"),}\\
<_N &:= \lesssim_N \setminus \gtrsim_N && \text{("properly more specific"),}\\
\leq_N &:= <_N \cup \{(X,X) \mid X \text{ is an argument}\} && \text{("more specific or equal"),}\\
\triangle_N &:= \left\{ (X,Y) \;\middle|\; \begin{array}{l} X, Y \text{ are arguments with} \\ X \not\lesssim_N Y \text{ and } X \not\gtrsim_N Y \end{array} \right\} && \text{("incomparable w.r.t. specificity").}
\end{aligned}$$

A *quasi-ordering* is a reflexive transitive relation. An *(irreflexive) ordering* is an irreflexive transitive relation. A *reflexive ordering* (also called: "partial ordering") is an anti-symmetric quasi-ordering. An *equivalence* is a symmetric quasi-ordering.

Corollary 1 *If $\lesssim_N$ is a quasi-ordering, then $\approx_N$ is an equivalence, $<_N$ is an ordering, and $\leq_N$ is a reflexive ordering.*

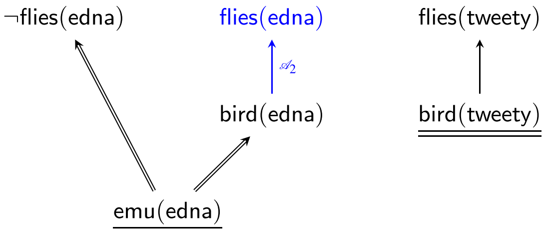
3 Motivating Examples

For ease of distinction, we will use the special symbol "$\leftarrow$" as a syntactic sugar in concrete examples of defeasible rules from Δ, instead of the symbol "$\Leftarrow$", which — in our concrete examples — will be used only in strict rules.

Moreover, in our graphical illustrations we will indicate membership in Π^{F} by *double underlining*.

Example 1 (Example 1 of [22])

$$\begin{aligned}
\Pi_1^{\mathrm{F}} &:= \left\{ \begin{array}{l} \mathsf{bird}(\mathsf{tweety}), \\ \mathsf{emu}(\mathsf{edna}) \end{array} \right\},\\
\Pi_1^{\mathrm{G}} &:= \left\{ \begin{array}{l} \mathsf{bird}(x) \Leftarrow \mathsf{emu}(x), \\ \neg\mathsf{flies}(x) \Leftarrow \mathsf{emu}(x) \end{array} \right\},\\
\Delta_1 &:= \{ \mathsf{flies}(x) \leftarrow \mathsf{bird}(x) \},\\
\mathcal{A}_2 &:= \{ \mathsf{flies}(\mathsf{edna}) \leftarrow \mathsf{bird}(\mathsf{edna}) \}.
\end{aligned}$$



We have $\begin{aligned} \mathfrak{T}_{\Pi_1} &= \{\mathsf{bird}(\mathsf{tweety}), \mathsf{emu}(\mathsf{edna}), \mathsf{bird}(\mathsf{edna}), \neg\mathsf{flies}(\mathsf{edna})\},\\ \mathfrak{T}_{\Pi_1 \cup \Delta_1} &= \{\mathsf{flies}(\mathsf{edna}), \mathsf{flies}(\mathsf{tweety})\} \cup \mathfrak{T}_{\Pi_1}. \end{aligned}$

It is intuitively clear that we prefer the argument $(\emptyset, \neg\mathsf{flies}(\mathsf{edna}))$ to the argument $(\mathcal{A}_2, \mathsf{flies}(\mathsf{edna}))$, simply because the former does not use any defeasible rules. We will further discuss this in Example 7. Q5

Let us see what happens to Example 1 if we are not so certain anymore that no emu can fly and turn the general rule $(\neg\mathsf{flies}(x) \Leftarrow \mathsf{emu}(x)) \in \Pi_1^{\mathrm{G}}$ into a defeasible one in the following example.

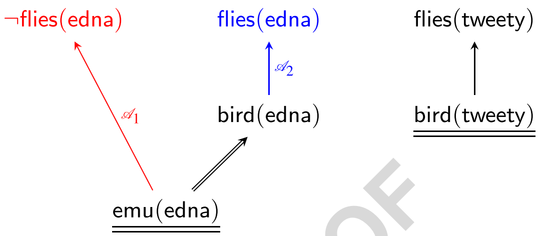
Example 2 (Example 2 of [22])

$$\Pi_2^{\mathrm{F}} := \left\{ \begin{array}{l} \mathsf{bird}(\mathsf{tweety}), \\ \mathsf{emu}(\mathsf{edna}) \end{array} \right\}, \quad \Pi_2^{\mathrm{G}} := \{ \mathsf{bird}(x) \Leftarrow \mathsf{emu}(x) \},$$

$$\Delta_2 := \left\{ \begin{array}{l} \neg\mathsf{flies}(x) \leftarrow \mathsf{emu}(x), \\ \mathsf{flies}(x) \leftarrow \mathsf{bird}(x) \end{array} \right\}.$$

$$\mathcal{A}_1 := \{ \neg\mathsf{flies}(\mathsf{edna}) \leftarrow \mathsf{emu}(\mathsf{edna}) \}.$$

$$\mathcal{A}_2 := \{ \mathsf{flies}(\mathsf{edna}) \leftarrow \mathsf{bird}(\mathsf{edna}) \}.$$



We have
$$\mathfrak{T}_{\Pi_2} = \{\mathsf{bird}(\mathsf{tweety}), \mathsf{emu}(\mathsf{edna}), \mathsf{bird}(\mathsf{edna})\},$$
$$\mathfrak{T}_{\Pi_2 \cup \Delta_2} = \{\neg\mathsf{flies}(\mathsf{edna}), \mathsf{flies}(\mathsf{edna}), \mathsf{flies}(\mathsf{tweety})\} \cup \mathfrak{T}_{\Pi_2}.$$

It is intuitively clear that we prefer the argument $(\mathcal{A}_1, \neg\mathsf{flies}(\mathsf{edna}))$ to the argument $(\mathcal{A}_2, \mathsf{flies}(\mathsf{edna}))$, simply because the defeasible derivation of the former is based on $\mathsf{emu}(\mathsf{edna})$, and because this is more specific than $\mathsf{bird}(\mathsf{edna})$, on which the derivation of the latter argument is based. We will further discuss this in Example 8.

Let us see what happens to Example 2 if we doubt that emus are birds.

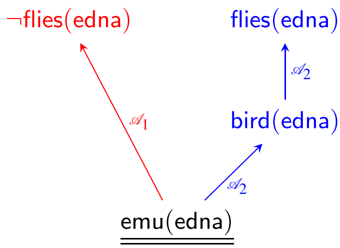
Example 3 (Renamed Subsystem of Example 3 of [22])

$$\Pi_3^{\mathrm{F}} := \{ \mathsf{emu}(\mathsf{edna}) \}, \qquad \Pi_3^{\mathrm{G}} := \emptyset,$$

$$\Delta_3 := \left\{ \begin{array}{l} \neg\mathsf{flies}(x) \leftarrow \mathsf{emu}(x), \\ \mathsf{flies}(x) \leftarrow \mathsf{bird}(x), \\ \mathsf{bird}(x) \leftarrow \mathsf{emu}(x) \end{array} \right\},$$

$$\mathcal{A}_1 := \{ \neg\mathsf{flies}(\mathsf{edna}) \leftarrow \mathsf{emu}(\mathsf{edna}) \}.$$

$$\mathcal{A}_2 := \left\{ \begin{array}{l} \mathsf{flies}(\mathsf{edna}) \leftarrow \mathsf{bird}(\mathsf{edna}), \\ \mathsf{bird}(\mathsf{edna}) \leftarrow \mathsf{emu}(\mathsf{edna}) \end{array} \right\}.$$



We have

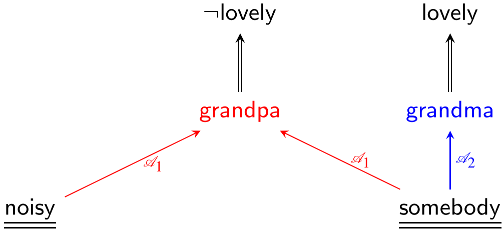
$$\mathfrak{T}_{\Pi_3} = \{\mathsf{emu}(\mathsf{edna})\}, \qquad \mathfrak{T}_{\Pi_3 \cup \Delta_3} = \{\mathsf{bird}(\mathsf{edna}), \mathsf{flies}(\mathsf{edna}), \neg\mathsf{flies}(\mathsf{edna})\} \cup \mathfrak{T}_{\P_3}.$$

Now it is not clear anymore whether we should prefer $(\mathcal{A}_1, \neg\mathsf{flies}(\mathsf{edna}))$ to $(\mathcal{A}_2, \mathsf{flies}(\mathsf{edna}))$. Both arguments are now based on $\mathsf{emu}(\mathsf{edna})$, but it is not clear whether the less specific $\mathsf{bird}(\mathsf{edna})$ — because it has dropped out of $\mathfrak{T}_{\Pi_3}$ now — can still be considered as a basis for $(\mathcal{A}_2, \mathsf{flies}(\mathsf{edna}))$. We will further discuss this in Example 9.

Now suppose that we have a lovely grandma and a grouchy and noisy grandpa, stay at their house and hear that somebody is coming into the house noisily, but cannot see yet who it is.

Example 4

$$
\begin{aligned}
\Pi_4^{\mathrm{F}} &:= \{\ \mathsf{somebody},\ \mathsf{noisy}\ \},\\
\Pi_4^{\mathrm{G}} &:= \left\{ \begin{array}{l} \mathsf{lovely} \Leftarrow \mathsf{grandma}, \\ \neg\mathsf{lovely} \Leftarrow \mathsf{grandpa} \end{array} \right\},\\
\Delta_4 &:= \mathscr{A}_1 \cup \mathscr{A}_2.\\
\mathscr{A}_1 &:= \{\ \mathsf{grandpa} \leftarrow \mathsf{somebody} \wedge \mathsf{noisy}\ \},\\
\mathscr{A}_2 &:= \{\ \mathsf{grandma} \leftarrow \mathsf{somebody}\ \}.
\end{aligned}
$$



Let us compare the specificity of the arguments $(\mathscr{A}_1, \neg\mathsf{lovely})$ and $(\mathscr{A}_2, \mathsf{lovely})$. We have

$$\mathfrak{T}_{\Pi_4} = \{\mathsf{somebody}, \mathsf{noisy}\}, \qquad \mathfrak{T}_{\Pi_4 \cup \Delta_4} = \{\mathsf{grandma}, \mathsf{grandpa}, \mathsf{lovely}, \neg\mathsf{lovely}\} \cup \mathfrak{T}_{\Pi_4}.$$

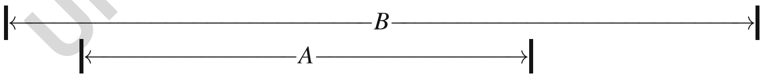
Now, because there is somebody who is noisy according to the current situation given by Π_4^{F}, it is probably grandpa because his characterization is more specific. Thus, it is intuitively clear that we would prefer $(\mathscr{A}_1, \neg\mathsf{lovely})$ as the more specific argument to $(\mathscr{A}_2, \mathsf{lovely})$. We will further discuss this in Example 10.

4 Toward an intuitive notion of specificity

4.1 The common-sense concept of specificity

It is part of general knowledge that a criterion is [properly] more specific than another one if the "class of candidates that satisfy it" is a [proper] subclass of that of the other one. Analogously — taking logical formulas as the criteria — a formula A is [properly] more specific than a formula B, if the model class of A is a [proper] subclass of the model class of B, i.e. if $A \models B$ [and $B \not\models A$].

If we consider a formula as a predicate on model-theoretic structures, its model class becomes the extension of this predicate. From this viewpoint, we can state $A \models B$ also as the syllogism "every A is B", and also as the following Lambert diagram [19, Dianoiologie, §§173–194].



4.2 Arguments as an abstraction

To enable a closer investigation of the critical parts of a defeasible derivation, we have to isolate the defeasible parts in the derivation. From a concrete derivation of a literal L, let us abstract the set $\mathscr{A}$ of the ground instances of the defeasible rules that are actually applied in the derivation, and form the pair $(\mathscr{A}, L)$, which we already called an *argument* in Definition 2 of Section 2.4.

4.3 The intuitive rôle of activation sets in the definition of specificity

If we want to classify a derivation with defeasible rules according to its specificity, then we have to isolate the defeasible part of the derivation and look at its input formulas, so that we can see how specific these input formulas are. The input formulas are the set of those literals on which the defeasible part of the derivation is based, called the *activation set* for the defeasible part of the derivation. In our framework of defeasible positive-conditional specification, the only relevant property of an activation set can be the conjunction of its literals which we can represent by the set itself.[2]

For instance, in Example 2 of Section 3, the argument $(\mathscr{A}_1, \neg\mathsf{flies}(\mathsf{edna}))$ is based only on the activation set $\{\mathsf{emu}(\mathsf{edna})\}$, whereas the argument $(\mathscr{A}_2, \mathsf{flies}(\mathsf{edna}))$ can also be based on the activation set $\{\mathsf{bird}(\mathsf{edna})\}$, or on the union of these sets.

Moreover, in Example 4 of Section 3, the argument $(\mathscr{A}_1, \neg\mathsf{lovely})$ is based only on the activation set $\{\mathsf{somebody}, \mathsf{noisy}\}$, whereas the argument $(\mathscr{A}_2, \mathsf{lovely})$ can also be based on the less specific activation set $\{\mathsf{somebody}\}$.

4.3.1 Modulo which theory are activation sets to be compared?

Because all literals of an activation set have been derived from the given specification, it does not make sense to compare activation sets w.r.t. the models of the entire specification. Indeed, only a comparison w.r.t. the models of a sub-specification can show any differences between them.

Therefore, we have to find out which parts of a specification $(\Pi^{\mathrm{F}}, \Pi^{\mathrm{G}}, \Delta)$ are to be excluded from the comparison of activation sets.

We want to have the *entire* set Π^{G} available for our comparison of activation sets, for the following reasons: The general and strict part Π^{G} of our specification represents the necessary and stable kernel of our rules, independent of the concrete situation under consideration given by Π^{F}, and independent of the uncertainty of our default rules Δ. Moreover, it is hardly meaningful to exclude any proper rule from Π^{G} (i.e. any rule from Π^{G} that is not just a literal); the technical reason for this will be given right at the beginning of Section 4.4.3.

We have to exclude Π^{F} from this comparison, however. This exclusion makes sense because the defeasible rules are typically default rules not written in particular for the given concrete situation that is formalized by Π^{F}. Moreover, as indicated before, the inclusion of Π^{F} would typically eliminate all differences between activation sets, such as it is the case in all examples of Section 3.

Finally, as we want to compare the defeasible parts of derivations, we should exclude the set Δ of the defeasible rules when we compare activation sets. Thus, on the one hand, all we can take into account from our specification is a subset of the general rules Π^{G}, and, on the other hand, we do not want to exclude any of these general rules.

All in all, we conclude that Π^{G} is that part of our specification modulo which activation sets are to be compared.

4.3.2 A first sketch of a notion of specificity

Very roughly speaking, if we have fewer activation sets for the defeasible part of a derivation, then these activation sets describe fewer models (i.e. their disjunction has fewer models), which again means that the defeasible part of the derivation is more specific. Accordingly, a first sketch of a notion of specificity can now be given as follows:

[2] A formal definition of an activation set is not needed here and would be harmful to intuition. Several different formal notions of activation sets will be found in Definition 7 of Section 6.1 and also in Definition 16 of Section 8.3.1.

An argument $(\mathscr{A}_1, L_1)$ is [properly] *more specific than* an argument $(\mathscr{A}_2, L_2)$ if, for each activation set H_1 for $(\mathscr{A}_1, L_1)$, there is an activation set $H_2 \subseteq \mathfrak{T}_{H_1 \cup \Pi^{\mathrm{G}}}$ for $(\mathscr{A}_2, L_2)$ [but not vice versa].

Note that this notion of specificity is preliminary, and that the notion of an activation set for an argument has not been properly defined yet.

4.4 Isolation of the defeasible parts of a derivation

If $(\mathscr{A}, L)$ is an argument (cf. Section 4.2), then there is a derivation of L which is based only on those instances of defeasible rules which are contained in $\mathscr{A}$. Such an argument ignores the concrete derivation, and therefore suits our model-theoretic intentions (cf. Section 1). With such an argument as an abstraction of a derivation, however, we lose the possibility to isolate the actual defeasible parts of the derivation. Such a loss is typical for abstractions in general; in our case, however, the discussion of this loss in Section 4.4.1 will turn out to be conceptually crucial and result in several different formal notions of activation sets.[3]

4.4.1 Isolation of actual defeasible parts in and-trees

Let us compare this set $\mathscr{A}$ with an *and-tree of the derivation*. Every node in such a tree is labeled with the conclusion of an instance of a rule, such that its children are labeled exactly with the elements of the conjunction in the condition of this instance.

Definition 6 (And-Tree)
Let $(\Pi^{\mathrm{F}}, \Pi^{\mathrm{G}}, \Delta)$ be a defeasible specification (cf. Section 2.3), and let L be a literal.

An *and-tree* T for L [and for the derivation of $\Phi \vdash \{L\}$] w.r.t. $(\Pi^{\mathrm{F}}, \Pi^{\mathrm{G}}, \Delta)$ is a finite, rooted tree, where every node is labeled with a literal, satisfying the following conditions:

1. The root node of T is labeled with L.
2. For each node N in T labeled with a literal L', there is a strict or defeasible rule $(L''_0 \Leftarrow L''_1 \wedge \ldots \wedge L''_k) \in \Pi \cup \Delta$, such that $L' = L''_0\sigma$ for some substitution σ [with $(L''_0\sigma \Leftarrow L''_1\sigma \wedge \ldots \wedge L''_k\sigma) \in \Phi$]. Moreover, the node N has exactly k child nodes, which are labeled with $L''_1\sigma, \ldots, L''_k\sigma$, respectively.

This standard and very simple formal notion of an and-tree is meant to capture a single derivation for a single argument. It must not be confused with the compact multi-graphs that come as a synopsis with our examples (such as the ones in Section 3).[4]

An isolation of the defeasible parts of an and-tree of the derivation may now proceed as follows:

– Starting from the root of the tree, we iteratively erase all applications of strict rules. This gives us a set of trees, each of which has the application of a defeasible rule at the root.
– Starting now from the leaves of these trees, we again erase all applications of strict rules. This gives us a set of trees with the following property holding for every node:

[3] See Definition 7 of Section 6.1 and also Definition 16 of Section 8.3.1.

[4] These sophisticated multi-graphs illustrate several derivations for several arguments in parallel, share sub-graphs, and may have =-edges between occurrences of the same literal L to represent alternative derivations of L (cf. Example 6 in Section 6.2 as well as Example 15 and 16 in Section 7.2). Because these synopses are redundant in all examples, we do not provide a formalization for these multi-graphs.

If *all* children of a node (if there are any) are leaves, then this node results from an application of a defeasible rule.

4.4.2 A first approximation of activation sets

In a first approximation, we may now take the activation set for the original derivation to be the set of all labels L of all leaves of all resulting trees, unless the literal L is an unconditional rule from $\mathscr{A}$.

The motivation for this notion of an activation set is that the conjunction of its literals is a weakest precondition for all defeasible parts of the concrete original derivation. If such a logically weakest precondition satisfies the specificity notion of Section 4.3.2 as an activation set for an argument $(\mathscr{A}_1, L_1)$ w.r.t. a second argument $(\mathscr{A}_2, L_2)$, then any other precondition for all defeasible parts of the given and-tree will satisfy this notion w.r.t. $(\mathscr{A}_2, L_2)$ a fortiori.[5]

4.4.3 Growth of the defeasible parts toward the leaves

Note that in the set of trees resulting from the procedure described at the end of Section 4.4.1, there may well have remained instances of rules from Π^G connecting a defeasible root application with the defeasible applications right at the leaves. Thus — to cover the whole defeasible part of the derivation in our abstraction — we have to consider the set $\mathscr{A} \cup \Pi^G$ instead of just the set $\mathscr{A}$.

More precisely, we have to include all proper rules (i.e. those with non-empty conditions) from Π^G, and may also include the literals in Π^G because they cannot do any harm.[6]

As a consequence, in the modeling via our abstraction $\mathscr{A}$, we cannot prevent the isolated defeasible sub-trees resulting from the procedure described in Section 4.4.1 from using the rules from Π^G to grow toward the root and toward the leaves again. Only the growth toward the leaves, however, can affect our activation sets (which are still taken to be the labels of all leaves of all resulting trees) and thereby our notion of specificity. Indeed, a growth toward the root can add to the conjunction of the given leaves only its super-conjunctions, which are irrelevant because of our focus on weakest preconditions (explained in Section 4.4.2).

Let us have a closer look at the effects of such a growth toward the leaves in the most simple case. In addition to a given activation set $\{\mathsf{Q}(\mathsf{a})\}$, in the presence of a general rule

$$\mathsf{Q}(x) \Leftarrow \mathsf{P}_0(x) \wedge \cdots \wedge \mathsf{P}_{n-1}(x)$$

from Π^G, we will also have to consider the activation set $\{\mathsf{P}_i(\mathsf{a}) | i \in \{0, \ldots, n-1\}\}$.

This has two effects, which we will discuss in Sections 4.4.4 and 4.4.5.

[5]Note that a further dissection of the isolated defeasible parts would not in general result in activation sets that can be inferred from the strict rules in Π. Where this inference is possible, however, a further dissection leads to the special notion of activation sets given in Definition 16 of Section 8.3.1.

[6]The need to include all proper rules and to exclude the literals from Π^F provides a motivation for simply defining Π^G to contain exactly the proper rules of Π, such as found in [27].

4.4.4 *First effect: simplified second sketch of a notion of specificity*

The first effect is that we immediately realize that every model of Π^{G} in the model class that is represented by the activation set $\{\mathsf{P}_i(\mathsf{a}) | i \in \{0,\ldots,n-1\}\}$ is also in the model class represented by the activation set $\{\mathsf{Q}(\mathsf{a})\}$.

Indeed, this growth toward the leaves will immediately add $\{\mathsf{P}_i(\mathsf{a}) | i \in \{0,\ldots,n-1\}\}$ as a further activation set for every argument with the activation set $\{\mathsf{Q}(\mathsf{a})\}$. By this effect it is just made explicit that an argument that can be based on the activation set $\{\mathsf{Q}(\mathsf{a})\}$ can also be based on the activation set $\{\mathsf{P}_i(\mathsf{a}) | i \in \{0,\ldots,n-1\}\}$. Thus — provided that there are no other activation sets — an argument that can be based on the activation set $\{\mathsf{Q}(\mathsf{a})\}$ is less or equivalently specific compared to any argument that can be based on $\{\mathsf{P}_i(\mathsf{a}) | i \in \{0,\ldots,n-1\}\}$.

Therefore — if we admit the effect of a growth toward the leaves on our activation sets — we may simplify[7] the comparison of activation sets in our first sketch of a notion of specificity of Section 4.3.2 as follows:

> An argument $(\mathscr{A}_1, L_1)$ is [properly] *more specific than* an argument $(\mathscr{A}_2, L_2)$ if, for each activation set H_1 for $(\mathscr{A}_1, L_1)$, this set H_1 is also an activation set for $(\mathscr{A}_2, L_2)$ [but not vice versa].

4.4.5 *Second effect: preference of the "more concise"*

The second effect, however, is that an argument $(\mathscr{A}_2, L_2)$ that gets along with $\{\mathsf{Q}(\mathsf{a})\}$ becomes even *properly* less specific than an argument $(\mathscr{A}_1, L_1)$ that actually requires $\{\mathsf{P}_i(\mathsf{a}) | i \in \{0,\ldots,n-1\}\}$. and does not get along with $\{\mathsf{Q}(\mathsf{a})\}$, simply because $(\mathscr{A}_2, L_2)$ has the additional activation set $\{\mathsf{Q}(\mathsf{a})\}$.

The resulting preference of $(\mathscr{A}_1, L_1)$ to $(\mathscr{A}_2, L_2)$ as being properly more specific is usually called *preference of the "more concise"*, cf. e.g. [27, p. 94], [13, p. 108]. Although — to the best of our knowledge — this notion has never been formally defined, roughly speaking it is — for an instantiated rule $\mathsf{Q}(\mathsf{a}) \Leftarrow \mathsf{P}_0(\mathsf{a}) \wedge \cdots \wedge \mathsf{P}_{n-1}(\mathsf{a})$ of the specification — the preference of an argument that gets along with the conclusion $\{\mathsf{Q}(\mathsf{a})\}$ of the instantiated rule as an activation set, instead of actually requiring the condition $\{\mathsf{P}_i(\mathsf{a}) | i \in \{0,\ldots,n-1\}\}$.

For instance, in Example 2 of Section 3, an argument that gets along with $\{\mathsf{bird(edna)}\}$ is properly less specific than one that actually requires $\{\mathsf{emu(edna)}\}$, in the sense that $\mathsf{emu(edna)}$ is more concise than $\mathsf{bird(edna)}$.

The problem now is that the statement $\mathsf{Q}(\mathsf{a}) \not\models \mathsf{P}_0(\mathsf{a}) \wedge \cdots \wedge \mathsf{P}_{n-1}(\mathsf{a})$ — which is required to justify this preference — is not explicitly given by the specification $(\Pi^{\mathrm{F}}, \Pi^{\mathrm{G}}, \Delta)$.

Nevertheless — if we do not just want to see it as a matter-of-fact property of notions of specificity in the style of Poole — we could justify the preference of the "more concise" by imposing the following best practice on positive-conditional specification:

If we write an implication in form of a rule

$$\mathsf{Q}(x) \Leftarrow \mathsf{P}_0(x) \wedge \cdots \wedge \mathsf{P}_{n-1}(x)$$

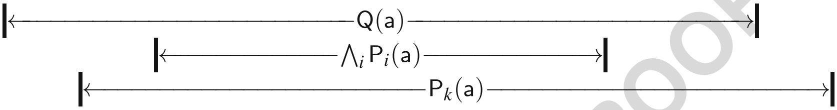
[7] Note that we have replaced here the option to choose some activation set $H_2 \subseteq \mathfrak{T}_{H_1 \cup \Pi^{\mathrm{G}}}$ of the first sketch with the restrictive determination $H_2 := H_1$. This simplifying restriction applies here for the following reason: If $H_2 \subseteq \mathfrak{T}_{H_1 \cup \Pi^{\mathrm{G}}}$ is an activation set for $(\mathscr{A}_2, L_2)$, then H_1 is an activation set for $(\mathscr{A}_2, L_2)$ as well, provided that we admit the first effect of a growth toward the leaves via Π^{G} on our activation sets.

JrnlID 10472_ArtID 9471_Proof#1 - 26/08/2015

into a positive-conditional specification Π of strict (i.e. non-defeasible) knowledge, and if we do not intend that the implication is proper in the sense that its converse does not hold in general, then we ought to specify the full equivalence by adding the rules $\mathsf{P}_i(x) \Leftarrow \mathsf{Q}(x)$ $(i \in \{0, \ldots, n-1\})$ to the specification.[8]

Under this best practice of specification, if we find such a rule without the specification of its full equivalence, then it is not intended to exclude models where Q holds for some object a, but not all of the P_i do. This means that if we find such a rule in the strict and general part Π^{G} of a specification, then it is reasonable to assume that the implication is proper w.r.t. the intuition captured in the defeasible rules in Δ.

As a consequence, it makes sense to consider a defeasible argument based on $\{\,\mathsf{P}_i(\mathsf{a}) | i \in \{0, \ldots, n-1\}\,\}$ to be properly more specific than an argument that can get along with $\mathsf{Q}(\mathsf{a})$.



Remark 5 (Justification for Preference of the "More Concise" Not Valid for Defeasible Rules)
Note that our justification for the preference of the "more concise" does not apply, however, if $\mathsf{Q}(x) \Leftarrow \mathsf{P}_0(x) \wedge \cdots \wedge \mathsf{P}_{n-1}(x)$ is a *defeasible* rule instead of a strict one, because we then have the following three problems when trying to justify preference of the "more concise":

- The implication given by the rule is not generally intended (otherwise the rule should be a strict one).
- Moreover, we cannot easily describe the actual instances to which the default rule is meant to apply (otherwise this more concrete description of the defeasible rule should be stated as strict rules).
- The direct treatment of a defeasible equivalence neither has to be appropriate as a default rule in the given situation, nor do we have any means to express a defeasible equivalence in the current setting.

Accordingly, there is, for instance, no clear reason to prefer the first argument of Example 3 in Section 3 to the second one. This will be discussed in more detail in Example 9.

[8]There is one exception to this justification, however, in the practice of *logic programming*: If $\mathsf{Q}(x) \Leftarrow \mathsf{P}_0(x) \wedge \cdots \wedge \mathsf{P}_{n-1}(x)$ is the only rule of the specification with Q as the predicate symbol of the conclusion, then it is standard in PROLOG to consider this implication as an implementation of a full equivalence defining the predicate Q.

This is different in our context of *positive-conditional specification* here, however, where we can add and ought to add the rules $\mathsf{P}_i(x) \Leftarrow \mathsf{Q}(x)$ $(i \in \{0, \ldots, n-1\})$ to our specification, simply because we are not concerned with the non-termination problem of logic programming resulting from such a specification of the full equivalence (cf. Section 2.1).

An alternative which is given also in logic programming is to omit the rule indicated above and to replace each occurrence of each $\mathsf{Q}(t)$ with $\mathsf{P}_0(t) \wedge \cdots \wedge \mathsf{P}_{n-1}(t)$, respectively.

Moreover, in the frequent case that several cases of the definition of a predicate are spread over several rules, the implications definitely tend to be proper also in logic programming, because, roughly speaking, the defined predicate is given as the proper disjunction of the conditions of the several rules.

4.4.6 *Preference of the "more precise"*

If we consider an argument requiring an activation set $\{\ \mathsf{P}_i(\mathsf{a}) | i \in \{0,\ldots,n-1\}\ \}$ to be *properly* more specific than an argument that gets along with a proper subset $\{\ \mathsf{P}_i(\mathsf{a}) | i \in I\}$ for some index set $I \subsetneq \{0,\ldots,n\}$, then the resulting preference is usually called *preference of the "more precise"*, cf. e.g. [27, p. 94], [13, p. 108]. An example for the preference of the "more precise" is Example 4 of Section 3.

There is, however, an exception from this preference to be observed, namely the case that we can actually derive the set from its subset with the help of Π^{G}. In this case, the above-mentioned growth toward the leaves with rules from Π^{G} again implements the approximation of the subclass relation among model classes via the one among activation sets.[9]

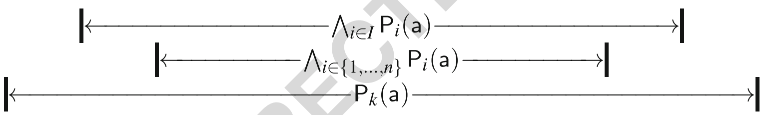
Apart from this exception, there is again a problem, namely that it is not the case that

$$\bigwedge_{i \in I} \mathsf{P}_i(\mathsf{a}) \not\models \bigwedge\nolimits_{i \in \{0,\ldots,n\}} \mathsf{P}_i(\mathsf{a})$$

would be explicitly given by the specification via $(\Pi^{\mathrm{F}}, \Pi^{\mathrm{G}}, \Delta)$.

Nevertheless — if we do not just want to see it as a matter-of-fact property of notions of specificity in the style of Poole — we could justify also the preference of the "more precise" by imposing the following best practice on positive-conditional specification:

If we want to exclude the above non-consequence, then we ought to specify, for each $j \in \{0,\ldots,n\} \setminus I$, a rule like $\mathsf{P}_j(x) \Leftarrow \bigwedge_{i \in I} \mathsf{P}_i(x)$.



4.4.7 *Conclusion on the preferences*

Let us finally point out that an acceptance of our justifications of the preferences of the "more concise" and the "more precise" is not at all a prerequisite for following our investigations on Poole's model-theoretic notion of specificity and our correction of this notion in the following sections.

5 Requirements specification of specificity in positive-conditional specification

With implicit reference to a defeasible specification $(\Pi^{\mathrm{F}}, \Pi^{\mathrm{G}}, \Delta)$ (cf. Section 2.3), let us designate Poole's relation of being more (or equivalently) specific by "$\precsim_{\mathrm{P1}}$". Here, "P1" stands for "Poole's original version".

The standard usage of the symbol "$\precsim$" is to denote a *quasi-ordering* (cf. Section 2.5). Instead of the symbol "$\precsim$", however, [22] uses the symbol "$\leq$". The standard usage of the symbol "$\leq$" is to denote a *reflexive ordering* (cf. Section 2.5). We cannot conclude from this, however, that Poole intended the additional property of anti-symmetry; indeed, we find

[9]This approximation was discussed in Section 4.4.4 and will be demonstrated in Example 18 of Section 7.

a concrete example specification in Poole [22] where the lack of anti-symmetry of $\lesssim_{\mathrm{P1}}$ is made explicit.[10]

The possible lack of anti-symmetry of quasi-orderings — i.e. that different arguments may have an equivalent specificity — cannot be a problem because any quasi-ordering $\lesssim_N$ immediately provides us with its equivalence $\approx_N$, its ordering $<_N$, and its reflexive ordering $\leq_N$ (cf. Corollary 1 of Section 2.5).

By contrast to the non-intended anti-symmetry, *transitivity* is obviously a *conditio sine qua non* for any useful notion of specificity. Indeed, if we have to make a quick choice among the three mutually exclusive actions Propose, Kiss, Smile, and if we already have an argument $(\mathscr{A}_2, \mathsf{Kiss})$ that is more specific than another argument $(\mathscr{A}_3, \mathsf{Smile})$, and if we come up with yet another argument $(\mathscr{A}_1, \mathsf{Propose})$ that is even more specific than $(\mathscr{A}_2, \mathsf{Kiss})$, then, by all means, $(\mathscr{A}_1, \mathsf{Propose})$ should be more specific than the argument $(\mathscr{A}_3, \mathsf{Smile})$ as well. It is obvious that a notion of specificity without transitivity could hardly be helpful in practice.

A further *conditio sine qua non* for any useful notion of specificity is that the conjunctive combination of respectively more specific arguments results in a more specific argument. Indeed, if a square is more specific than a rectangle and a circle is more specific than an ellipse, then a square inscribed into a circle should be more specific than a rectangle inscribed into an ellipse. This property is called *monotonicity of conjunction*, which we will discuss in Section 7.1. Already in [22], we find an example[11] where $\lesssim_{\mathrm{P1}}$ violates this monotonicity property of the conjunction, which is described there as "seemingly unintuitive".[12]

Further intricacies of computing Poole's specificity in concrete examples are described in [27],[13] which will make it hard to implement $\lesssim_{\mathrm{P1}}$ or its minor corrections as efficiently as required in the practice of answer computation and SLD-resolution w.r.t. positive-conditional specification s.

6 Formalizations of specificity

6.1 Activation sets

A derivation from the leaves to the root can now be split into three phases of derivation of literals from literals. This splitting follows the discussion in Section 4.4.1 on how to isolate the defeasible parts of a derivation (phase 2) from strict parts that may occur toward the root (phase 3) and toward the leaves (phase 1):

(phase 1) First we derive the literals that provide the basis for specificity considerations. In our approach we derive the set $\mathfrak{T}_\Pi$ here. Poole takes the set $\mathfrak{T}_{\Pi\cup\Delta}$ instead.

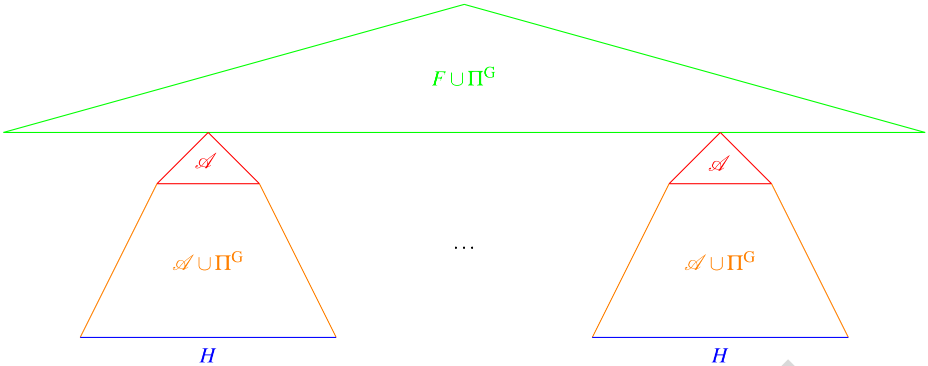
(phase 2) On the basis of

[10] Here we refer to the last three sentences of Section 3.2 on Page 145 of [22].

[11] Here we refer to Example 6 of [22, Section 3.5, p. 146], see our Example 12 in Section 7.1.

[12] See our Example 12 in Section 7.1 and the references there.

[13] Here we refer to Section 3.2ff of [27], where it is demonstrated that, for deciding Poole's specificity relation (actually $\lesssim_{\mathrm{P2}}$ instead of $\lesssim_{\mathrm{P1}}$, but this does not make any difference here) for two input arguments, we sometimes have to consider even those defeasible rules which are not part of any of these arguments. See also our Example 15 in Section 7.2.



	$H \subseteq \ldots$	$F = \ldots$
Poole's Approach (P1, P2, P3)	$\mathfrak{T}_{\Pi^{F} \cup \Pi^{G} \cup \Delta}$	H
Our Approach (CP1, CP2)	$\mathfrak{T}_{\Pi^{F} \cup \Pi^{G}}$	Π^{F}

Fig. 1 And-tree with phases 1, 2, 3[16,17,18] Q6

- a subset H of the literals derived in phase 1,
- the first item $\mathscr{A}$ of a given argument $(\mathscr{A}, L)$, and
- the general rules Π^{G},

we derive a further set of literals $\mathfrak{L}$: $H \cup \mathscr{A} \cup \Pi^{G} \vdash \mathfrak{L}$.

(phase 3) Finally, on the basis of $\mathfrak{L}$, the literal of the given argument $(\mathscr{A}, L)$ is derived: $\mathfrak{L} \cup \Pi \vdash \{L\}$.

In Poole's approach, phase 3 is empty and we simply have $\mathfrak{L} = \{L\}$. In our approach, however, it is admitted to use the facts from Π^{F} in phase 3, in addition to the general rules from Π^{G}, which were already admitted in phase 2.

With implicit reference to our sets $\Pi = \Pi^{F} \cup \Pi^{G}$ and Δ, the phases 2 and 3 can be more easily expressed with the help of the following notions.

Definition 7 ([Minimal] [Simplified] Activation Set)
Let $\mathscr{A}$ be a set of ground instances of rules from Δ, and let L be a literal.
H is a *simplified activation set for* $(\mathscr{A}, L)$ if $L \in \mathfrak{T}_{H \cup \mathscr{A} \cup \Pi^{G}}$.
H is an *activation set for* $(\mathscr{A}, L)$ if $L \in \mathfrak{T}_{\mathfrak{L} \cup \Pi}$ for some $\mathfrak{L} \subseteq \mathfrak{T}_{H \cup \mathscr{A} \cup \Pi^{G}}$.
H is a *minimal* [*simplified*] *activation set for* $(\mathscr{A}, L)$ if H is an [simplified] activation set for $(\mathscr{A}, L)$, but no proper subset of H is an [simplified] activation set for $(\mathscr{A}, L)$.

Corollary 2 *Let $\mathscr{A}$ be a set of ground instances of rules from Δ, and let L be a literal. Every simplified activation set for $(\mathscr{A}, L)$ is an activation set for $(\mathscr{A}, L)$.*

Roughly speaking, an argument is now more (or equivalently) specific than another one if each of its activation sets is also an activation set for the other argument. Note that this follows the simplified second sketch of a notion of specificity displayed in Section 4.4.4, not the first one displayed in Section 4.3.2.

Activation sets that are not simplified differ from simplified ones by the admission of facts from Π^{F} (in addition to the general rules Π^{G}) after the defeasible part of the derivation is completed.[14]

Our introduction of activation sets that are not simplified is a conceptually important correction of Poole's approach: It must be admitted to use the facts besides the general rules in a purely strict derivation that is based on literals resulting from completed defeasible arguments, simply because the defeasible parts of a derivation (as isolated in Section 4.4.1) should not get more specific by the later use of additional facts that do not provide input to the defeasible parts.[15] Note that the difference between simplified and non-simplified activation sets typically occurs in real applications, but — except Example 16 in Section 7.2 — not in our toy examples of Section 7, which mainly exemplify the differences in phase 1.

6.2 Poole's specificity relation P1 and its minor corrections P2, P3

In this section we will define the binary relations $\lesssim_{\mathrm{P1}}$, $\lesssim_{\mathrm{P2}}$, $\lesssim_{\mathrm{P3}}$ of "being more or equivalently specific according to David Poole" with implicit reference to our sets of facts and of general and defeasible rules (i.e. to Π^{F}, Π^{G}, and Δ, respectively).

The relation $\lesssim_{\mathrm{P1}}$ of the following definition is precisely Poole's original relation $\geq$ as defined at the bottom of the left column on Page 145 of [22]. See Section 5 for our reasons to write "$\gtrsim$" instead of "$\geq$" as a first change. Moreover, as a second change required by mathematical standards, we have replaced the symbol "$\gtrsim$" with the symbol "$\lesssim$" (such that the smaller argument becomes the more specific one), so that the relevant well-foundedness becomes the one of its ordering $<$ instead of the reverse $>$.

Definition 8 ($\lesssim_{\mathrm{P1}}$: David Poole's Original Specificity)
$(\mathcal{A}_1, L_1) \lesssim_{\mathrm{P1}} (\mathcal{A}_2, L_2)$ if $(\mathcal{A}_1, L_1)$ and $(\mathcal{A}_2, L_2)$ are arguments, and if, for every $H \subseteq \mathfrak{T}_{\Pi\cup\Delta}$ that is a simplified activation set for $(\mathcal{A}_1, L_1)$ but not a simplified activation set for $(\mathcal{A}_2, L_1)$, H is also a simplified activation set for $(\mathcal{A}_2, L_2)$.

The relation $\lesssim_{\mathrm{P2}}$ of the following definition is the relation $\succeq$ of [27, Definition 10, p. 94] (attributed to [22]). Moreover, the relation $>_{\mathrm{spec}}$ of [26, Definition 2.12, p. 132] (attributed to [22] as well) is the relation $<_{\mathrm{P2}} := \lesssim_{\mathrm{P2}} \setminus \gtrsim_{\mathrm{P2}}$.

[14] This can be seen in Example 16 of Section 7, and in Example 19 of Section 8.2.2. See also the variable F in Fig. 1.

[15] We do not further discuss this obviously appropriate correction here and leave the construction of examples that make the conceptual necessity of this correction intuitively clear as an exercise. Hint: Have a look at the proof of Theorem 3 in Section 6.5. Then present two different sets of strict rules with equal derivability, where only one needs the facts in phase 3 and where the additional specificity gained by these facts violates the intuition.

[16] Look at Note 30 of Example 15 in Section 7.2 to see that it may really matter for the definition of P1, P2, P3 that we do *not* have $F \subseteq \mathfrak{T}_{\Pi^{\mathrm{F}}\cup\Pi^{\mathrm{G}}}$ in general in Poole's approach.

[17] Although we do *not* have $H \subseteq \Pi^{\mathrm{F}}$ in general in our approach, the replacement of Π^{F} with H in this table would result in fewer derivable roots for our approach, simply because we always have $\mathfrak{T}_{H\cup\Pi^{\mathrm{G}}} \subseteq \mathfrak{T}_{\Pi^{\mathrm{F}}\cup\Pi^{\mathrm{G}}}$ in our approach.

[18] From the leaves to the root: phase 1 (H), phase 2 (sub-trees of the defeasible parts of a derivation, with explicit defeasible root steps), phase 3 (root sub-tree). For Poole's approach, however, the root sub-tree is still part of phase 2, whereas phase 3 is empty.

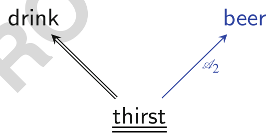
Definition 9 ($\lesssim_{\mathrm{P2}}$: Standard Version of David Poole's Specificity)
$(\mathscr{A}_1, L_1) \lesssim_{\mathrm{P2}} (\mathscr{A}_2, L_2)$ if $(\mathscr{A}_1, L_1)$ and $(\mathscr{A}_2, L_2)$ are arguments, and if, for every $H \subseteq \mathfrak{T}_{\Pi\cup\Delta}$ that is a simplified activation set for $(\mathscr{A}_1, L_1)$ but not a simplified activation set for $(\emptyset, L_1)$, H is also a simplified activation set for $(\mathscr{A}_2, L_2)$.

The only change in Definition 9 as compared to Definition 8 is that "$(\mathscr{A}_2, L_1)$" is replaced with "$(\emptyset, L_1)$". We did not yet encounter any example where any difference results from this correction toward "$(\emptyset, L_1)$", which is standard in the publications of the last two decades and which is intuitively more appropriate in the sense of a weight or measure function.

The relations $\lesssim_{\mathrm{P1}}$ and $\lesssim_{\mathrm{P2}}$ were not meant to compare arguments for literals that do not need any defeasible rules — or at least they do not show an intuitive behavior on such arguments, as shown in Example 5.

Example 5 (Minor Flaw of $\lesssim_{\mathrm{P1}}$ and $\lesssim_{\mathrm{P2}}$)

$$\Pi_5^{\mathrm{F}} := \{\mathsf{thirst}\}, \quad \Pi_5^{\mathrm{G}} := \{\mathsf{drink} \Leftarrow \mathsf{thirst}\},$$
$$\Delta_5 := \mathscr{A}_2.$$
$$\mathscr{A}_2 := \{\, \mathsf{beer} \leftarrow \mathsf{thirst} \,\}.$$



Let us compare the specificity of the arguments $(\mathscr{A}_2, \mathsf{beer})$ and $(\emptyset, \mathsf{drink})$, meaning that we should have a beer or else an arbitrary drink at our own choice, respectively.

We have $\mathfrak{T}_{\Pi_5} = \{\mathsf{thirst}, \mathsf{drink}\}$, $\mathfrak{T}_{\Pi_5\cup\Delta_5} = \{\mathsf{beer}\} \cup \mathfrak{T}_{\Pi_5}$.

We have $(\mathscr{A}_2, \mathsf{beer}) \lesssim_{\mathrm{P2}} (\emptyset, \mathsf{drink})$ because for every $H \subseteq \mathfrak{T}_{\Pi_5\cup\Delta_5}$ that is a simplified activation set for $(\mathscr{A}_2, \mathsf{beer})$, but not a simplified activation set for $(\emptyset, \mathsf{beer})$, we have $\mathsf{thirst} \in H$, so H is a simplified activation set also for $(\emptyset, \mathsf{drink})$.

We have $(\emptyset, \mathsf{drink}) \lesssim_{\mathrm{P2}} (\mathscr{A}_2, \mathsf{beer})$ because there cannot be a simplified activation set for $(\emptyset, \mathsf{drink})$ that is not a simplified activation set for $(\emptyset, \mathsf{drink})$.

All in all, we get[19] $(\mathscr{A}_2, \mathsf{beer}) \approx_{\mathrm{P2}} (\emptyset, \mathsf{drink})$, although $(\emptyset, \mathsf{drink})$ should be strictly preferred to $(\mathscr{A}_2, \mathsf{beer})$ according to intuition, simply because an argument that does not require any defeasible rules should always be strictly preferred to a comparable argument that does actually require defeasible rules.

To overcome this minor flaw, which consists in the inconvenience of not in general preferring a non-defeasible argument to a comparable defeasible one, we finally add an implication as an additional requirement in Definition 10. This implication guarantees that no argument that requires defeasible rules can be more or equivalently specific than an argument that does not require any defeasible rules at all.

Definition 10 ($\lesssim_{\mathrm{P3}}$: Rather Unflawed Version of David Poole's Specificity)
$(\mathscr{A}_1, L_1) \lesssim_{\mathrm{P3}} (\mathscr{A}_2, L_2)$ if $(\mathscr{A}_1, L_1)$ and $(\mathscr{A}_2, L_2)$ are arguments, $L_2 \in \mathfrak{T}_{\Pi}$ implies

[19] Note that by Corollary 4, we will get $(\mathscr{A}_2, \mathsf{beer}) \approx_{\mathrm{P1}} (\emptyset, \mathsf{drink})$ as well. Moreover, note that this problem does not occur in the similar Example 1 of Section 3.

$L_1 \in \mathfrak{T}_\Pi$, and if, for every $H \subseteq \mathfrak{T}_{\Pi\cup\Delta}$ that is a [minimal][20] simplified activation set for $(\mathscr{A}_1, L_1)$ but not a simplified activation set for $(\emptyset, L_1)$, H is also a simplified activation set for $(\mathscr{A}_2, L_2)$.

Corollary 3 *If $(\mathscr{A}_1, L_1)$, $(\mathscr{A}_2, L_2)$ are arguments with $\mathscr{A}_1 \subseteq \mathscr{A}_2$, then any of the following conditions is sufficient for $(\mathscr{A}_1, L_1) \precsim_{P3} (\mathscr{A}_2, L_2)$:*

1. $L_1 = L_2$.
2. $L_2 \in \mathfrak{T}_\Pi \implies L_1 \in \mathfrak{T}_\Pi$ *and* $\{L_1\} \cup \mathscr{A}_2 \cup \Pi^{\mathrm{G}} \vdash \{L_2\}$,
3. $\mathscr{A}_1 = \emptyset$ *(which implies $L_1 \in \mathfrak{T}_\Pi$ by Definition 5).*[21]

As every simplified activation set that passes the condition of Definition 8 also passes the one of Definitions 9 and 10,, we get the following corollary of these three definitions.

Corollary 4 $\precsim_{P3} \subseteq \precsim_{P2} \subseteq \precsim_{P1}$.

By Corollaries 3 and 4, $\precsim_{\mathrm{P1}}$, $\precsim_{\mathrm{P2}}$, and $\precsim_{\mathrm{P3}}$ are reflexive relations on arguments, but — as we will show in Example 6 and state in Theorem 1 — not quasi-orderings in general.

Example 6 (Counterexample to the Transitivities: "Choose one action!")
Suppose you meet the sexy girl Jo in a lift for a very short time, you smile at her, and she smiles back with a head akimbo. Since smiling, kissing, and proposing are mutually exclusive actions of your mouth, you have to make up your mind quickly what to do next, depending on your current level of boldness.[22]

$$\Pi_6^{\mathrm{F}} := \{\mathsf{Bold}, \mathsf{HAkimbo(Jo)}, \mathsf{Smiles(Jo)}, \mathsf{Sexy(Jo)}\},$$

$$\Pi_6^{\mathrm{G}} := \{\mathsf{Kiss} \Leftarrow \mathsf{Promising}(G)\},$$

$$\Delta_6 := \left\{\begin{array}{l} \mathsf{Smile} \leftarrow \mathsf{Sexy}(G), \\ \mathsf{Kiss} \leftarrow \mathsf{Bold} \wedge \mathsf{Smiles}(G) \wedge \mathsf{Sexy}(G), \\ \mathsf{Promising}(G) \leftarrow \mathsf{HAkimbo}(G) \wedge \mathsf{Smiles}(G) \wedge \mathsf{Sexy}(G), \\ \mathsf{Propose} \leftarrow \mathsf{Promising}(G) \wedge \mathsf{Bold} \end{array}\right\}.$$

$$\mathscr{A}_1 := \left\{\begin{array}{l} \mathsf{Promising(Jo)} \leftarrow \mathsf{HAkimbo(Jo)} \wedge \mathsf{Smiles(Jo)} \wedge \mathsf{Sexy(Jo)} \\ \mathsf{Propose} \leftarrow \mathsf{Promising(Jo)} \wedge \mathsf{Bold} \end{array}\right\},$$

$$\mathscr{A}_2 := \{\mathsf{Kiss} \leftarrow \mathsf{Bold} \wedge \mathsf{Smiles(Jo)} \wedge \mathsf{Sexy(Jo)}\},$$

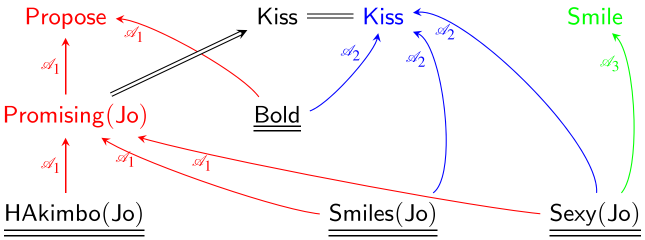
$$\mathscr{A}_3 := \{\mathsf{Smile} \leftarrow \mathsf{Sexy(Jo)}\}.$$

Compare the specificity of the arguments $(\mathscr{A}_1, \mathsf{Propose})$, $(\mathscr{A}_2, \mathsf{Kiss})$, $(\mathscr{A}_3, \mathsf{Smile})$!

[20] Note that the omission of the optional restriction to *minimal* simplified activation sets for $(\mathscr{A}_1, L_1)$ in Definition 10 has no effect on the extension of the defined notion, simply because the additional non-minimal simplified activation sets $(\mathscr{A}_1, L_1)$ will then be simplified activation sets for $(\mathscr{A}_2, L_2)$ *a fortiori*.

[21] Exercise: Find a counterexample, however, for the conjecture that $L_1 \in \mathfrak{T}_\Pi$ implies $(\mathscr{A}, L_1) \precsim_{\mathrm{P3}} (\mathscr{A}, L_2)$.

[22] The nullary predicate Bold could actually be removed from all rules and facts of this example, which would still remain a counterexample to the transitivities; to the contrary, it would even improve its status by becoming a *minimal* counterexample. A renaming of the resulting minimal counterexample was presented as Example 5.8 in [34, 35].



Lemma 1 *There are*

- *a specification* $(\Pi^{\mathrm{F}}_6, \Pi^{\mathrm{G}}_6, \Delta_6)$ *without any negative literals (i.e., a fortiori,* $\Pi^{\mathrm{F}}_6 \cup \Pi^{\mathrm{G}}_6 \cup \Delta_6$ *is non-contradictory), and*
- *minimal arguments* $(\mathscr{A}_1, L_1)$, $(\mathscr{A}_2, L_2)$, $(\mathscr{A}_3, L_3)$,

such that $(\mathscr{A}_1, L_1) \lesssim_{P3} (\mathscr{A}_2, L_2) \lesssim_{P3} (\mathscr{A}_3, L_3) \not\gtrsim_{P1} (\mathscr{A}_1, L_1)$ *and* $(\mathscr{A}_1, L_1) \not\gtrsim_{P1} (\mathscr{A}_2, L_2) \not\gtrsim_{P1} (\mathscr{A}_3, L_3)$.

Proof of Lemma 1 Looking at Example 6, we see that only the quasi-ordering properties in the last two lines of Lemma 1 are non-trivial. We have

$$\mathfrak{T}_{\Pi_6} = \{\mathsf{Bold}, \mathsf{HAkimbo(Jo)}, \mathsf{Smiles(Jo)}, \mathsf{Sexy(Jo)}\},$$
$$\mathfrak{T}_{\Pi_6 \cup \Delta_6} = \{\mathsf{Promising(Jo)}, \mathsf{Propose}, \mathsf{Kiss}, \mathsf{Smile}\} \cup \mathfrak{T}_{\Pi_6}.$$

Thus, regarding the arguments $(\mathscr{A}_1, \mathsf{Propose})$, $(\mathscr{A}_2, \mathsf{Kiss})$, $(\mathscr{A}_3, \mathsf{Smile})$, the implication added in Definition 10 as compared to Definitions 8 and 9 is always satisfied, simply because its condition is always false.

$(\mathscr{A}_3, \mathsf{Smile}) \not\gtrsim_{\mathrm{P1}} (\mathscr{A}_1, \mathsf{Propose}) \lesssim_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{Kiss})$: The minimal simplified activation sets for $(\mathscr{A}_1, \mathsf{Propose})$ that are subsets of $\mathfrak{T}_{\Pi_6 \cup \Delta_6}$ and no simplified activation sets for $(\emptyset, \mathsf{Propose})$ (or, without any difference, no simplified activation sets for $(\mathscr{A}_3, \mathsf{Propose})$) are $\{\mathsf{Bold}, \mathsf{HAkimbo(Jo)}, \mathsf{Smiles(Jo)}, \mathsf{Sexy(Jo)}\}$ and $\{\mathsf{Bold}, \mathsf{Promising(Jo)}\}$, which are simplified activation sets for $(\mathscr{A}_2, \mathsf{Kiss})$ — but $\{\mathsf{Bold}, \mathsf{Promising(Jo)}\}$ is no simplified activation set for $(\mathscr{A}_3, \mathsf{Smile})$.

$(\mathscr{A}_1, \mathsf{Propose}) \not\gtrsim_{\mathrm{P1}} (\mathscr{A}_2, \mathsf{Kiss}) \lesssim_{\mathrm{P3}} (\mathscr{A}_3, \mathsf{Smile})$: The only simplified activation set for $(\mathscr{A}_2, \mathsf{Kiss})$ that is a subset of $\mathfrak{T}_{\Pi_6 \cup \Delta_6}$ and no simplified activation set for $(\emptyset, \mathsf{Kiss})$ (such as $\{\mathsf{Promising(Jo)}\}$) (or, without any difference, no simplified activation set for $(\mathscr{A}_1, \mathsf{Kiss})$) is $\{\mathsf{Bold}, \mathsf{Smiles(Jo)}, \mathsf{Sexy(Jo)}\}$, which is a simplified activation set for $(\mathscr{A}_3, \mathsf{Smile})$, but not for $(\mathscr{A}_1, \mathsf{Propose})$.

$(\mathscr{A}_2, \mathsf{Kiss}) \not\gtrsim_{\mathrm{P1}} (\mathscr{A}_3, \mathsf{Smile})$: The only minimal simplified activation set for $(\mathscr{A}_3, \mathsf{Smile})$ that is a subset of $\mathfrak{T}_{\Pi_6 \cup \Delta_6}$ and no simplified activation set for $(\mathscr{A}_2, \mathsf{Smile})$ is $\{\mathsf{Sexy(Jo)}\}$, which is not a simplified activation set for $(\mathscr{A}_2, \mathsf{Kiss})$.

□

6.3 Main negative result: not transitive!

The relations stated in Lemma 1 hold not only for the given indices, but — by Corollary 4 — actually for all of P1, P2, P3; and so we immediately get:

JrnlID 10472_ArtID 9471_Proof#1 - 26/08/2015

Theorem 1
There is a specification $(\Pi^{\mathrm{F}}_6, \Pi^{\mathrm{G}}_6, \Delta_6)$, *such that* $\Pi^{\mathrm{F}}_6 \cup \Pi^{\mathrm{G}}_6 \cup \Delta_6$ *is non-contradictory, but none of* $\lesssim_{P1}$, $\lesssim_{P2}$, $\lesssim_{P3}$, $<_{P1}$, $<_{P2}$, $<_{P3}$ *is transitive. Moreover, the counterexamples to the transitivity of all these relations can be restricted to minimal arguments.*

As a consequence of Theorem 1, the respective relations in [22, 27], and [26] are not transitive. This means that these relations are not quasi-orderings, let alone reflexive orderings.

This consequence is immediate for the relation $\geq$ at the bottom of the left column on Page 145 of [22]. Moreover, note that the consequence does not depend on the contentious question on whether our interpretation of the negation symbol $\neg$ essentially differs from its interpretation in [22]. Indeed, our counterexample to transitivity occurs in the negation-free definite-rule fragment of Poole's original language.

Moreover, this consequence is also immediate for the relation $\succeq$ [27, Definition 10, p. 94] and for the relation $>_{\mathrm{spec}}$ [26, Definition 2.12, p.132], simply because we can replace $\succeq$ and $>_{\mathrm{spec}}$ with $\lesssim_{\mathrm{P2}}$ and $<_{\mathrm{P2}}$ in the context of Example 6, respectively.

Although transitivity of these relations is strongly suggested by the special choice of their symbols and seems to be taken for granted in general, we found an actual statement of such a transitivity only for the relation $\sqsupseteq$ of [26, Definition 2.22, p.134], namely in "Lemma 2.23" [26, p. 134].[23]

Finally, note that those readers who do not see a proper conflict in our counterexample just should add to Example 6 some general rules such as Execute $\Leftarrow$ Kiss, Execute $\Leftarrow$ Smile, $\neg$Execute $\Leftarrow$ Propose, say to model the situation in one of the areas of today's planet Earth where an unmarried woman who raises the wish to smile or kiss has to be executed.

6.4 Our novel specificity ordering CP1

In the previous section, we have seen that *minor corrections* of Poole's original relation P1 (such as P2, P3) do not cure the (up to our finding of Example 6) hidden or even denied deficiency of these relations, namely their lack of transitivity. Our true motivation for a *major correction* of P3 was not this formal deficiency, but actually an informal one, namely that it failed to get sufficiently close to human intuition, which will become clear in Section 7.

For these reasons, we now define our major correction of Poole's specificity — the binary relation $\lesssim_{\mathrm{CP1}}$ — with implicit reference to our sets of facts and of general and defeasible rules (i.e. to Π^{F}, Π^{G}, and Δ, respectively) as follows.

[23] According to the rules of good scientific and historiographic practice, we pinpoint the violation of this "lemma" now as follows. Non-transitivity of $\sqsupseteq$ follows here immediately from the non-transitivity of the relation $\geq_{\mathrm{spec}}$ of Definition 2.15, which, however, is not identical to the above-mentioned relation $\succeq$, but actually a subset of $\succeq$, because it is defined via a peculiar additional equivalence $\approx_{\mathrm{spec}}$ introduced in Definition 2.14, [26, p. 132], namely via $\geq_{\mathrm{spec}} := >_{\mathrm{spec}} \cup \approx_{\mathrm{spec}}$ [26, Definition 2.15, p.132f.]. Directly from Definition 2.14 of [26], we get $\approx_{\mathrm{spec}} \subseteq \approx_{\mathrm{P2}}$. Thus, by Corollary 4, we get $\geq_{\mathrm{spec}} \subseteq \lesssim_{\mathrm{P2}} \subseteq \lesssim_{\mathrm{P1}}$; and so (recollecting $<_{\mathrm{P2}} \subseteq >_{\mathrm{spec}} \subseteq \geq_{\mathrm{spec}}$) the result

$$(\mathscr{A}_1, L_1) <_{\mathrm{P2}} (\mathscr{A}_2, L_2) <_{\mathrm{P2}} (\mathscr{A}_3 L_3) \not\gtrsim_{\mathrm{P1}} (\mathscr{A}_1, L_1)$$

of Lemma 1 gives us the following counterexample to transitivity:

$$(\mathscr{A}_1, L_1) \geq_{\mathrm{spec}} (\mathscr{A}_2, L_2) \geq_{\mathrm{spec}} (\mathscr{A}_3 L_3) \not\leq_{\mathrm{spec}} (\mathscr{A}_1, L_1).$$

Definition 11 ($\precsim_{\mathrm{CP1}}$: 1st Version of our Specificity Relation)
$(\mathscr{A}_1, L_1) \precsim_{\mathrm{CP1}} (\mathscr{A}_2, L_2)$ if $(\mathscr{A}_1, L_1)$ and $(\mathscr{A}_2, L_2)$ are arguments, and we have

1. $L_1 \in \mathfrak{T}_\Pi$ or
2. $L_2 \notin \mathfrak{T}_\Pi$ and every $H \subseteq \mathfrak{T}_\Pi$ that is an [minimal][24] activation set for $(\mathscr{A}_1, L_1)$ is also an activation set for $(\mathscr{A}_2, L_2)$.

Corollary 5 *If* $(\mathscr{A}_1, L_1)$, $(\mathscr{A}_2, L_2)$ *are arguments with* $\mathscr{A}_1 \subseteq \mathscr{A}_2$*, then any of the following conditions is sufficient for* $(\mathscr{A}_1, L_1) \precsim_{\mathit{CP1}} (\mathscr{A}_2, L_2)$*:*

1. $L_1 = L_2$.
2. $L_2 \in \mathfrak{T}_\Pi \implies L_1 \in \mathfrak{T}_\Pi$ *and* $\{L_1\} \cup \Pi \vdash \{L_2\}$.[25]
3. $L_1 \in \mathfrak{T}_\Pi$ *(which is implied by* $\mathscr{A}_1 = \emptyset$ *by Definition 5).*

The crucial change in Definition 11 as compared to Definition 10 is *not* the technically required emphasis it puts on the case "$L_1 \in \mathfrak{T}_\Pi$", which will be discussed in Remark 6 of Section 6.6. The crucial changes actually are

(A) the replacement of "$H \subseteq \mathfrak{T}_{\Pi \cup \Delta}$" with "$H \subseteq \mathfrak{T}_\Pi$" (as explained already in phase 1 of Section 6.1), and the thereby enabled
(B) omission of the previously technically required,[26] but unintuitive negative condition on derivability (of the form "but not a simplified activation set for $(\emptyset, L_1)$").

An additional minor change, which we have already discussed in Section 6.1, is the one from simplified activation sets to (non-simplified) activation sets.

Theorem 2 $\precsim_{\mathit{CP1}}$ *is a quasi-ordering on arguments.*

Proof of Theorem 2
$\precsim_{\mathrm{CP1}}$ is a reflexive relation on arguments because of Corollary 5.

To show transitivity, let us assume $(\mathscr{A}_1, L_1) \precsim_{\mathrm{CP1}} (\mathscr{A}_2, L_2)$ and $(\mathscr{A}_2, L_2) \precsim_{\mathrm{CP1}} (\mathscr{A}_3, L_3)$. According to Definition 11, because of $(\mathscr{A}_1, L_1) \precsim_{\mathrm{CP1}} (\mathscr{A}_2, L_2)$, we have $L_1 \in \mathfrak{T}_\Pi$ — and then immediately the desired $(\mathscr{A}_1, L_1) \precsim_{\mathrm{CP1}} (\mathscr{A}_3, L_3)$ — or we have $L_2 \notin \mathfrak{T}_\Pi$ and every $H \subseteq \mathfrak{T}_\Pi$ that is an activation set for $(\mathscr{A}_1, L_1)$ is also an activation set for $(\mathscr{A}_2, L_2)$. The latter case excludes the first option in Definition 11 as a justification for $(\mathscr{A}_2, L_2) \precsim_{\mathrm{CP1}} (\mathscr{A}_3, L_3)$, and thus we have $L_3 \notin \mathfrak{T}_\Pi$ and every $H \subseteq \mathfrak{T}_\Pi$ that is an activation set for $(\mathscr{A}_2, L_2)$ is also an activation set for $(\mathscr{A}_3, L_3)$. All in all, we get that every $H \subseteq \mathfrak{T}_\Pi$ that is an activation set for $(\mathscr{A}_1, L_1)$ is also an activation set for $(\mathscr{A}_3, L_3)$. Thus, we get the desired $(\mathscr{A}_1, L_1) \precsim_{\mathrm{CP1}} (\mathscr{A}_3, L_3)$ also in this case. □

[24]Note that the omission of the optional restriction to *minimal* activation sets for $(\mathscr{A}_1, L_1)$ in Definition 11 has no effect on the extension of the defined notion, simply because the additional non-minimal activation sets for $(\mathscr{A}_1, L_1)$ will then be activation sets for $(\mathscr{A}_2, L_2)$ *a fortiori*.

[25]Note that, in general — contrary to Corollary 3(2) — $\mathscr{A}_2$ must not participate in the derivation of L_2 from L_1, say in the form that there is a set of literals $\mathfrak{L}$ with $\{L_1\} \cup \mathscr{A}_2 \cup \Pi^{\mathrm{G}} \vdash \mathfrak{L}$ and $\mathfrak{L} \cup \Pi \vdash \{L_2\}$, because rules from Π^{F} may have participated in the derivation of L_1 from an activation set. The source of this difference between P3 and CP1 is the replacement of simplified activation sets in Definition 10 with (non-simplified) activation sets in Definition 11.

[26]See the discussion in Example 10 in Section 6.6 on why this condition is technically required for P1, P2, and P3.

JrnlID 10472_ArtID 9471_Proof#1 - 26/08/2015

Obviously, an argument is ranked by $\lesssim_{\mathrm{CP1}}$ firstly on whether its literal is in $\mathfrak{T}_\Pi$, and, if not, secondly on the set of its activation sets, which is an element of the power set of the power set of $\mathfrak{T}_\Pi$. So we get:

Corollary 6 *If $\mathfrak{T}_\Pi$ is finite, then $<_{CP1}$ is well-founded.*

6.5 Relation between the specificity relations P3 and CP1

Theorem 3 *Let $\Pi^{<2}$ be the set of rules from Π that are unconditional or have exactly one literal in the conjunction of their condition.*

Let $\Pi^{\geq 2}$ be the set of rules from Π with more than one literal in their condition.

$\lesssim_{P3} \subseteq \lesssim_{CP1}$ holds if one (or more) of the following conditions hold:

1. *For every $H \subseteq \mathfrak{T}_\Pi$ and for every set $\mathscr{A}$ of ground instances of rules from Δ, and for $\mathfrak{L} := \mathfrak{T}_{H \cup \mathscr{A} \cup \Pi^{\mathrm{G}}}$, we have $\mathfrak{T}_{\mathfrak{L} \cup \Pi} \subseteq \mathfrak{L} \cup \mathfrak{T}_\Pi$.*
2. *For each instance $L \Leftarrow L'_0 \wedge \ldots \wedge L'_{n+1}$ of each rule in $\Pi^{\geq 2}$ with $L \notin \mathfrak{T}_{\Pi^{<2}}$, we have $L'_j \notin \mathfrak{T}_{\Pi^{<2}}$ for all $j \in \{0, \ldots, n+1\}$.*
3. *For each instance $L \Leftarrow L'_0 \wedge \ldots \wedge L'_{n+1}$ of each rule in $\Pi^{\geq 2}$, we have $L'_j \notin \mathfrak{T}_\Pi$ for all $j \in \{0, \ldots, n+1\}$.*
4. *We have $\Pi^{\geq 2} = \emptyset$.*

Note that if we had improved $\lesssim_{\mathrm{P3}}$ only w.r.t. phase 1 of Section 6.1, but not w.r.t. phase 3 in addition, then Theorem 3 would not require any condition at all. (See the proof!) This means that a condition becomes necessary by our correction of simplified activation sets to non-simplified ones, but not because of the major changes (A) and (B) of Section 6.4.

Proof of Theorem 3

First let us show that condition 2 implies condition 1. To this end, let $H \subseteq \mathfrak{T}_\Pi$, let $\mathscr{A}$ be a set of ground instances of rules from Δ, and set $\mathfrak{L} := \mathfrak{T}_{H \cup \mathscr{A} \cup \Pi^{\mathrm{G}}}$. For an *argumentum ad absurdum*, let us assume $\mathfrak{T}_{\mathfrak{L} \cup \Pi} \not\subseteq \mathfrak{L} \cup \mathfrak{T}_\Pi$. Because of $\Pi^{\mathrm{F}} \subseteq \mathfrak{T}_{\Pi^{<2}}$, we have $\mathfrak{L} \cup \Pi = \mathfrak{L} \cup \Pi^{\mathrm{F}} \cup \Pi^{\mathrm{G}} \subseteq \mathfrak{L} \cup \mathfrak{T}_{\Pi^{<2}} \cup \Pi^{\mathrm{G}}$, and thus $\mathfrak{T}_{\mathfrak{L} \cup \Pi} \subseteq \mathfrak{T}_{\mathfrak{L} \cup \mathfrak{T}_{\Pi^{<2}} \cup \Pi^{\mathrm{G}}}$, and thus $\mathfrak{T}_{\mathfrak{L} \cup \mathfrak{T}_{\Pi^{<2}} \cup \Pi^{\mathrm{G}}} \not\subseteq \mathfrak{L} \cup \mathfrak{T}_{\Pi^{<2}}$ (because otherwise $\mathfrak{T}_{\mathfrak{L} \cup \Pi} \subseteq \mathfrak{T}_{\mathfrak{L} \cup \mathfrak{T}_{\Pi^{<2}} \cup \Pi^{\mathrm{G}}} \subseteq \mathfrak{L} \cup \mathfrak{T}_{\Pi^{<2}} \subseteq \mathfrak{L} \cup \mathfrak{T}_\Pi$). Now $\mathfrak{L}$ is closed under Π^{G} by definition. Moreover, $\mathfrak{T}_{\Pi^{<2}}$ is closed under $\Pi^{<2}$ by definition and under $\Pi^{\geq 2}$ by condition 2. Because both of the sets of literals $\mathfrak{L}$ and $\mathfrak{T}_{\Pi^{<2}}$ are closed under Π^{G} — but nevertheless their union is not closed under Π^{G} according to $\mathfrak{T}_{\mathfrak{L} \cup \mathfrak{T}_{\Pi^{<2}} \cup \Pi^{\mathrm{G}}} \not\subseteq \mathfrak{L} \cup \mathfrak{T}_{\Pi^{<2}}$ — there must be an inference step *essentially based on both sets in parallel*. More precisely, this means that there must be an instance $L \Leftarrow L'_1 \wedge \ldots \wedge L'_n$ of a rule from Π^{G} with $L \notin \mathfrak{L} \cup \mathfrak{T}_{\Pi^{<2}}$, and some $i, j \in \{1, \ldots, n\}$ with $L'_i \in \mathfrak{L} \setminus \mathfrak{T}_{\Pi^{<2}}$ and $L'_j \in \mathfrak{T}_{\Pi^{<2}} \setminus \mathfrak{L}$. Then $L \Leftarrow L'_1 \wedge \ldots \wedge L'_n$ must actually be an instance of a rule from $\Pi^{\geq 2}$, and $L \notin \mathfrak{T}_{\Pi^{<2}}$, but $L'_j \in \mathfrak{T}_{\Pi^{<2}}$ in contradiction to condition 2.

As condition 2 implies condition 1, condition 3 trivially implies condition 2, and condition 4 trivially implies condition 3, it now suffices to show the claim that $(\mathscr{A}_1, L_1) \lesssim_{\mathrm{CP1}} (\mathscr{A}_2, L_2)$ holds under condition 1 and the assumption of $(\mathscr{A}_1, L_1) \lesssim_{\mathrm{P3}} (\mathscr{A}_2, L_2)$. By this assumption, $(\mathscr{A}_1, L_1)$ and $(\mathscr{A}_2, L_2)$ are arguments and $L_2 \in \mathfrak{T}_\Pi$ implies $L_1 \in \mathfrak{T}_\Pi$. If $L_1 \in \mathfrak{T}_\Pi$ holds, then our claim holds as well. Otherwise, we have $L_1, L_2 \notin \mathfrak{T}_\Pi$, and it suffices to show the sub-claim that H is an activation set for $(\mathscr{A}_2, L_2)$ under the additional sub-assumption that $H \subseteq \mathfrak{T}_{\Pi}$ is an activation set for $(\mathscr{A}_1, L_1)$. Under the sub-assumption we also have $H \subseteq \mathfrak{T}_{\Pi\cup\Delta}$ because of $\mathfrak{T}_{\Pi} \subseteq \mathfrak{T}_{\Pi\cup\Delta}$, and, for $\mathfrak{L} := \mathfrak{T}_{H\cup\mathscr{A}_1\cup\Pi^{\mathrm{G}}}$, we have $L_1 \in \mathfrak{T}_{\mathfrak{L}\cup\Pi}$, and then, by condition 1, $L_1 \in \mathfrak{L} \cup \mathfrak{T}_{\Pi}$. Then, by our current case of $L_1, L_2 \notin \mathfrak{T}_{\Pi}$, we have $L_1 \in \mathfrak{L}$. Thus, H is a *simplified* activation set for $(\mathscr{A}_1, L_1)$.

Let us now provide an *argumentum ad absurdum* for the assumption that H is a simplified activation set also for $(\emptyset, L_1)$: Then we would have $L_1 \in \mathfrak{T}_{H\cup\Pi^{\mathrm{G}}}$, and because of $H \subseteq \mathfrak{T}_{\Pi}$ and $\Pi^{\mathrm{G}} \subseteq \Pi$ we get $L_1 \in \mathfrak{T}_{\mathfrak{T}_{\Pi}\cup\Pi} = \mathfrak{T}_{\Pi}$ — a contradiction to our current case of $L_1, L_2 \notin \mathfrak{T}_{\Pi}$. All in all, by our initial assumption, H must now be a simplified activation set for $(\mathscr{A}_2, L_2)$ and, *a fortiori* by Corollary 2, an activation set for $(\mathscr{A}_2, L_2)$, as was to be shown for our only remaining sub-claim. □

6.6 Checking up the previous examples

With the help of Theorem 3, we can now analyze the examples of Section 3, and also check how our relation CP1 behaves in case of our counterexample to transitivity. Note that condition 4 of Theorem 3 is satisfied for all of these examples.

Example 7 (*continuing Example 1 of Section 3*)
We have $(\mathscr{A}_2, \mathsf{flies}(\mathsf{edna})) \not\lesssim_{\mathrm{CP1}} (\emptyset, \neg\mathsf{flies}(\mathsf{edna}))$ because $\mathsf{flies}(\mathsf{edna}) \notin \mathfrak{T}_{\Pi_1}$ and $\neg\mathsf{flies}(\mathsf{edna}) \in \mathfrak{T}_{\Pi_1}$.

We have $(\emptyset, \neg\mathsf{flies}(\mathsf{edna})) \lesssim_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{flies}(\mathsf{edna}))$ by Corollary 3(3). Q7

All in all, by Theorem 3, we get $(\emptyset, \neg\mathsf{flies}(\mathsf{edna})) <_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{flies}(\mathsf{edna}))$ and $(\emptyset, \neg\mathsf{flies}(\mathsf{edna})) <_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{flies}(\mathsf{edna}))$.

Remark 6 One may ask why we did not define an additional quasi-ordering, say $\lesssim_{\mathrm{CP0}}$, simply by replacing the two conditions of Definition 11 with the single condition

> "$L_2 \in \mathfrak{T}_{\Pi}$ implies $L_1 \in \mathfrak{T}_{\Pi}$, and every $H \subseteq \mathfrak{T}_{\Pi}$ that is an [minimal] activation set for $(\mathscr{A}_1, L_1)$ is also an activation set for $(\mathscr{A}_2, L_2)$."

This would be more in the style of Definition 10 for $\lesssim_{\mathrm{P3}}$, and would also avoid the singular behavior of the first alternative condition of Definition 11, and so offer continuity advantages.[27] Moreover, for $\lesssim_{\mathrm{CP0}}$ instead of $\lesssim_{\mathrm{CP1}}$, items 1 and 2 (but not item 3) of Corollary 5 still hold, as well as Theorem 2 and its Corollary 6. Furthermore, we get $\lesssim_{\mathrm{CP0}} \subseteq \lesssim_{\mathrm{CP1}}$. It is fatal for $\lesssim_{\mathrm{CP0}}$, however, that this subset relation may be proper. For instance, $\lesssim_{\mathrm{CP0}}$ does not in general satisfy Theorem 3. Even worse, $\lesssim_{\mathrm{CP0}}$ does not show the proper behavior of $\lesssim_{\mathrm{CP1}}$ in Example 1 of Section 3, as discussed in Example 7 of Section 6.6:

We get $(\emptyset, \neg\mathsf{flies}(\mathsf{edna})) \bigtriangleup_{\mathrm{CP0}} (\mathscr{A}_2, \mathsf{flies}(\mathsf{edna}))$ instead of

$$(\emptyset, \neg\mathsf{flies}(\mathsf{edna})) <_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{flies}(\mathsf{edna})).$$

This can be seen by considering the activation set $\emptyset$ for $(\emptyset, \neg\mathsf{flies}(\mathsf{edna}))$, which is not an activation set for $(\mathscr{A}_2, \mathsf{flies}(\mathsf{edna}))$.

Such a behavior is obviously unacceptable in practice, and so we do not think that it makes sense to consider $\lesssim_{\mathrm{CP0}}$ any further.

Example 8 (*continuing Example 2 of Section 3*)
We have $(\mathscr{A}_2, \mathsf{flies}(\mathsf{edna})) \not\lesssim_{\mathrm{CP1}} (\mathscr{A}_1, \neg\mathsf{flies}(\mathsf{edna}))$ because $\mathsf{flies}(\mathsf{edna}) \notin \mathfrak{T}_{\Pi_2}$ and

[27] Cf. the discussion of such a continuity advantage in Section 7.1 for the monotonicity w.r.t. conjunction.

because $\{\mathsf{bird(edna)}\} \subseteq \mathfrak{T}_{\Pi_2}$ is an activation set for $(\mathscr{A}_2, \mathsf{flies(edna)})$, but not for $(\mathscr{A}_1, \neg\mathsf{flies(edna)})$.

We have $(\mathscr{A}_1, \neg\mathsf{flies(edna)}) \lesssim_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{flies(edna)})$, because $\mathsf{flies(edna)} \notin \mathfrak{T}_{\Pi_2}$ and because, if $H \subseteq \mathfrak{T}_{\Pi_2 \cup \Delta_2}$ is a simplified activation set for $(\mathscr{A}_1, \neg\mathsf{flies(edna)})$, but not for $(\emptyset, \neg\mathsf{flies(edna)})$, then we have $\mathsf{emu(edna)} \in H$, and thus H is a simplified activation set also for $(\mathscr{A}_2, \mathsf{flies(edna)})$.

All in all, by Theorem 3, we get $(\mathscr{A}_1, \neg\mathsf{flies(edna)}) <_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{flies(edna)})$

$$\text{and}(\mathscr{A}_1, \neg\mathsf{flies(edna)}) <_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{flies(edna)}).$$

Example 9 *(continuing Example 3 of Section 3)*
We have $(\mathscr{A}_2, \mathsf{flies(edna)}) \lesssim_{\mathrm{CP1}} (\mathscr{A}_1, \neg\mathsf{flies(edna)})$ because $\neg\mathsf{flies(edna)} \notin \mathfrak{T}_{\Pi_3}$ and, for every activation set $H \subseteq \mathfrak{T}_{\Pi_3}$ for $(\mathscr{A}_2, \mathsf{flies(edna)})$, we get $\mathsf{emu(edna)} \in H$, and so H is an activation set also for $(\mathscr{A}_1, \neg\mathsf{flies(edna)})$.

Nevertheless, we have $(\mathscr{A}_2, \mathsf{flies(edna)}) \not\lesssim_{\mathrm{P3}} (\mathscr{A}_1, \neg\mathsf{flies(edna)})$, because $\{\mathsf{bird(edna)}\} \subseteq \mathfrak{T}_{\Pi_3 \cup \Delta_3}$ is a simplified activation set for $(\mathscr{A}_2, \mathsf{flies(edna)})$, but neither for $(\emptyset, \mathsf{flies(edna)})$, nor for $(\mathscr{A}_1, \neg\mathsf{flies(edna)})$.

We have $(\mathscr{A}_1, \neg\mathsf{flies(edna)}) \lesssim_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{flies(edna)})$, because of $\mathsf{flies(edna)} \notin \mathfrak{T}_{\Pi_3}$ and because, if $H \subseteq \mathfrak{T}_{\Pi_3 \cup \Delta_3}$ is a simplified activation set for $(\mathscr{A}_1, \neg\mathsf{flies(edna)})$, but not for $(\emptyset, \neg\mathsf{flies(edna)})$, then we have $\mathsf{emu(edna)} \in H$ and thus H is a simplified activation set also for $(\mathscr{A}_2, \mathsf{flies(edna)})$.

All in all, by Theorem 3, we get $(\mathscr{A}_1, \neg\mathsf{flies(edna)}) \approx_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{flies(edna)})$

$$\text{and}(\mathscr{A}_1, \neg\mathsf{flies(edna)}) <_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{flies(edna)}).$$

From a conceptual point of view, we have to ask ourselves, whether we would like the two *defeasible* rule instances in $\mathscr{A}_2 = \{\mathsf{flies(edna)} \leftarrow \mathsf{bird(edna)}, \mathsf{bird(edna)} \leftarrow \mathsf{emu(edna)}\}$ to reduce the specificity of $(\mathscr{A}_2, \mathsf{flies(edna)})$ as compared to a system that seems equivalent for the given argument for $\mathsf{flies(edna)}$, namely the argument $(\{\mathsf{flies(edna)} \leftarrow \mathsf{emu(edna)}\}, \mathsf{flies(edna)})$.

Does the specificity of a defeasible reasoning step really reduce if we introduce intermediate literals (such as $\mathsf{bird(edna)}$ between $\mathsf{flies(edna)}$ and $\mathsf{emu(edna)}$)?

According to human intuition, this question has a negative answer, as we have already explained in Remark 5 at the end of Section 4.4.5.[28]

Example 10 *(continuing Example 4 of Section 3)*
We have $(\mathscr{A}_2, \mathsf{lovely}) \not\lesssim_{\mathrm{CP1}} (\mathscr{A}_1, \neg\mathsf{lovely})$ because $\mathsf{lovely} \notin \mathfrak{T}_{\Pi_4}$ and because $\{\mathsf{somebody}\} \subseteq \mathfrak{T}_{\Pi_4}$ is an activation set for $(\mathscr{A}_2, \mathsf{lovely})$, but not for $(\mathscr{A}_1, \neg\mathsf{lovely})$.

We have $(\mathscr{A}_1, \neg\mathsf{lovely}) \lesssim_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{lovely})$ because of $\mathsf{lovely} \notin \mathfrak{T}_{\Pi_4}$ and because, if $H \subseteq \mathfrak{T}_{\Pi_4 \cup \Delta_4}$ is a simplified activation set for $(\mathscr{A}_1, \neg\mathsf{lovely})$, but not for $(\emptyset, \neg\mathsf{lovely})$, then we have $\{\mathsf{somebody}, \mathsf{noisy}\} \subseteq H$, and so H is also a simplified activation set for $(\mathscr{A}_2, \mathsf{lovely})$.

All in all, by Theorem 3, we get $(\mathscr{A}_1, \neg\mathsf{lovely}) <_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{lovely})$

$$\text{and } (\mathscr{A}_1, \neg\mathsf{lovely}) <_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{lovely}).$$

[28] Moreover, Examples 12 and 13 will exhibit a strong reason to deny this question: the requirement of monotonicity w.r.t. conjunction. Furthermore, see Examples 14 for another example that makes even clearer why defeasible rules should be considered for their global semantic effect instead of their syntactic fine structure.

Note that we can nicely see here that the condition that H is not a simplified activation set for $(\emptyset, \neg\mathsf{lovely})$ is relevant in Definition 10. Without this condition we would have to consider the simplified activation set $\{\mathsf{grandpa}\}$ for $(\mathscr{A}_1, \neg\mathsf{lovely})$, which is not an activation set for $(\mathscr{A}_2, \mathsf{lovely})$; and so, contrary to our intuition, $(\mathscr{A}_1, \neg\mathsf{lovely})$ would not be more specific than $(\mathscr{A}_2, \mathsf{lovely})$ w.r.t. $\lesssim_{\mathrm{P3}}$ anymore.

Example 11 (*continuing Example 6 of Section 6.2*)
The following holds for our specification of Example 6 by Lemma 1 and Corollary 4:

$$(\mathscr{A}_1, \mathsf{Propose}) <_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{Kiss}) <_{\mathrm{P3}} (\mathscr{A}_3, \mathsf{Smile}) \not\gtrsim_{\mathrm{P3}} (\mathscr{A}_1, \mathsf{Propose}).$$

For our corrected relation CP1 we have:

$$(\mathscr{A}_1, \mathsf{Propose}) <_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{Kiss}) <_{\mathrm{CP1}} (\mathscr{A}_3, \mathsf{Smile}) >_{\mathrm{CP1}} (\mathscr{A}_1, \mathsf{Propose})$$

simply because the trouble-making set $\{\mathsf{Bold}, \mathsf{Promising(Jo)}\}$ is not to be considered here. Indeed, this set is not a subset of $\mathfrak{T}_{\Pi_6}$. The checking of the details is left to the reader. Note that, because of Lemma 1, Theorem 3, Theorem 2, and Corollary 1, all that is actually left to show is $(\mathscr{A}_1, \mathsf{Propose}) \not\gtrsim_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{Kiss}) \not\gtrsim_{\mathrm{CP1}} (\mathscr{A}_3, \mathsf{Smile})$.

7 Putting specificity to test w.r.t. human intuition

Before we will go on with further conceptual material and efficiency considerations in Section 8, let us put our two main notions of specificity — as formalized in the two binary relations $\lesssim_{\mathrm{P3}}$ and $\lesssim_{\mathrm{CP1}}$ — to test w.r.t. our changed phase 1 of Section 6.1 in a series of further examples.

Note that we can freely draw the consequence $\lesssim_{\mathrm{P3}} \subseteq \lesssim_{\mathrm{CP1}}$ of Theorem 3 because at least one[29] of its conditions is satisfied in all the following examples except Example 16, which is the only example in Section 7 with an activation set that actually is not a simplified one.

Besides freely applying Theorem 3 — to enable the reader to make his own selection of interesting examples — we are pretty explicit in all of the following examples.

7.1 Monotonicity of the specificity relations w.r.t. conjunction

Monotonicity w.r.t. conjunction is the following property for a binary relation R on arguments:

$$\begin{array}{lll} \text{In case of} & (\mathscr{A}_1^i, L_1^i)\, R\, (\mathscr{A}_2^i, L_2^i) & \text{for } i \in \{1, 2\}, \\ \text{we always have} & (\mathscr{A}_1^1 \cup \mathscr{A}_1^2, L_1')\, R\, (\mathscr{A}_2^1 \cup \mathscr{A}_2^2, L_2') & \end{array}$$

[29] Condition 4 of Theorem 3 is satisfied for Examples 2, 3, 4, and 18. Condition 3 (but not condition 4) is satisfied for Examples 12, 13, 14, 15 and 17.

for fresh constant literals L'_j with rules $L'_j \Leftarrow L^1_j \wedge L^2_j$ added to the general rules Π^{G} ($j \in \{1, 2\}$). In this case, we will call $(\mathcal{A}^1_j \cup \mathcal{A}^2_j, L'_1)$ the *conjunction* of the arguments $(\mathcal{A}^1_j, L^1_j)$ and $(\mathcal{A}^2_j, L^2_j)$.

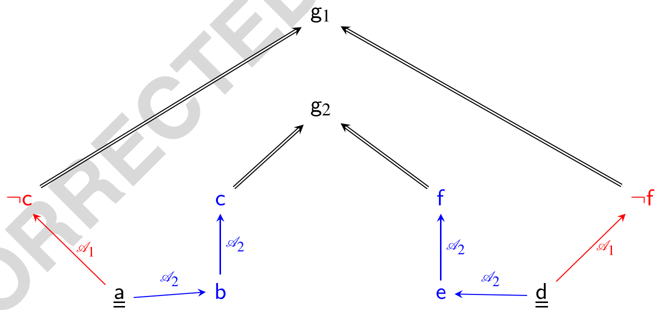
This property is obviously given for $\lesssim_{\mathrm{CP1}}$ in case of $L^1_1, L^2_1 \in \mathfrak{T}_\Pi$ (which implies $L'_1 \in \mathfrak{T}_\Pi$) and also in case of $L^1_1, L^2_1 \notin \mathfrak{T}_\Pi$ (where we get $L^1_2, L^2_2, L'_1, L'_2 \notin \mathfrak{T}_\Pi$). Note that the latter case — where both arguments are defeasible — is certainly the most important one.

For the remaining borderline case of $L^i_1 \notin \mathfrak{T}_\Pi \ni L^{3-i}_1$ (for some $i \in \{1, 2\}$), however, monotonicity cannot be expected in general for $\lesssim_{\mathrm{CP1}}$, simply because then we get $L'_1 \notin \mathfrak{T}_\Pi$, but do not necessarily have any activation set for L^{3-i}_2. This non-monotonicity, however, is part and parcel of our decision to prefer arguments whose literals are elements of $\mathfrak{T}_\Pi$, as expressed in item 1 of Definition 11 of Section 6.4. As explained in Remark 6 of Section 6.6, there does not seem to be an alternative to this technically required preference.

For $\lesssim_{\mathrm{P1}}$, however, monotonicity is not even given for the case we just realized to be the most important one. This was already noted in [22], using the following example.

Example 12 (*Example 6 of* [22])

$$\Pi^{\mathrm{F}}_{12} := \{\mathsf{a},\ \mathsf{d}\},$$
$$\Pi^{\mathrm{G}}_{12} := \left\{ \begin{array}{l} \mathsf{g}_1 \Leftarrow \neg\mathsf{c} \wedge \neg\mathsf{f}, \\ \mathsf{g}_2 \Leftarrow \mathsf{c} \wedge \mathsf{f} \end{array} \right\},$$
$$\Delta_{12} := \mathcal{A}_1 \cup \mathcal{A}_2.$$
$$\mathcal{A}_1 := \left\{ \begin{array}{l} \neg\mathsf{c} \leftarrow \mathsf{a}, \\ \neg\mathsf{f} \leftarrow \mathsf{d} \end{array} \right\}.$$
$$\mathcal{A}_2 := \left\{ \begin{array}{l} \mathsf{b} \leftarrow \mathsf{a}, \\ \mathsf{c} \leftarrow \mathsf{b}, \\ \mathsf{e} \leftarrow \mathsf{d}, \\ \mathsf{f} \leftarrow \mathsf{e} \end{array} \right\}.$$



Let us compare the specificity of the arguments $(\mathcal{A}_1, \mathsf{g}_1)$ and $(\mathcal{A}_2, \mathsf{g}_2)$.

We have $(\mathcal{A}_1, \mathsf{g}_1) \approx_{\mathrm{CP1}} (\mathcal{A}_2, \mathsf{g}_2)$ because $H \subseteq \mathfrak{T}_{\Pi_{12}} = \{\mathsf{a}, \mathsf{d}\}$ is an activation set for $(\mathcal{A}_i, \mathsf{g}_i)$ if and only if $H = \{\mathsf{a}, \mathsf{d}\}$.

We have $(\mathcal{A}_1, \mathsf{g}_1) \vartriangle_{\mathrm{P3}} (\mathcal{A}_2, \mathsf{g}_2)$ for the following reasons: $\{\mathsf{a}, \neg\mathsf{f}\} \subseteq \mathfrak{T}_{\Pi_{12} \cup \Delta_{12}}$ is a simplified activation set for $(\mathcal{A}_1, \mathsf{g}_1)$, but neither for $(\emptyset, \mathsf{g}_1)$, nor for $(\mathcal{A}_2, \mathsf{g}_2)$. $\{\mathsf{a}, \mathsf{f}\} \subseteq \mathfrak{T}_{\Pi_{12} \cup \Delta_{12}}$ is a simplified activation set for $(\mathcal{A}_2, \mathsf{g}_2)$, but neither for $(\emptyset, \mathsf{g}_2)$, nor for $(\mathcal{A}_1, \mathsf{g}_1)$.

Poole [22] considers the same result for $\lesssim_{\mathrm{P1}}$ as for $\lesssim_{\mathrm{P3}}$ to be "seemingly unintuitive", because, as we have seen for the isomorphic sub-specification in Example 3 of Section 3, we have both $(\mathcal{A}_1, \neg\mathsf{c}) <_{\mathrm{P3}} (\mathcal{A}_2, \mathsf{c})$ and $(\mathcal{A}_1, \neg\mathsf{f}) <_{\mathrm{P3}} (\mathcal{A}_2, \mathsf{f})$.

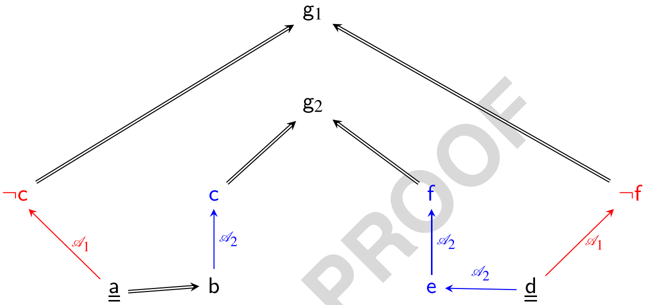
Indeed, as already listed as an essential requirement in Section 5, the conjunction of two respectively more specific arguments should be more specific.

On the other hand, considering $\lesssim_{\text{CP1}}$ instead of $\lesssim_{\text{P3}}$, the conjunctions of two respective arguments that are pairwise equivalently specific are equivalently specific — exactly as one intuitively expects. Indeed, from the isomorphic sub-specifications in Example 3, we know that $(\mathscr{A}_1, \neg\mathsf{c}) \approx_{\text{CP1}} (\mathscr{A}_2, \mathsf{c})$ and $(\mathscr{A}_1, \neg\mathsf{f}) \approx_{\text{CP1}} (\mathscr{A}_2, \mathsf{f})$.

By turning the defeasible rule $\mathsf{b} \leftarrow \mathsf{a}$ of Example 12 into a strict general rule, we obtain the following example.

Example 13 (*1st Variation of Example 12*)

$$\Pi^{\mathrm{F}}_{13} := \{\mathsf{a},\ \mathsf{d}\},$$
$$\Pi^{\mathrm{G}}_{13} := \left\{\begin{array}{l}\mathsf{g}_1 \Leftarrow \neg\mathsf{c}\wedge\neg\mathsf{f},\\ \mathsf{g}_2 \Leftarrow \mathsf{c}\wedge\mathsf{f},\\ \mathsf{b} \Leftarrow \mathsf{a}\end{array}\right\},$$
$$\Delta_{13} := \mathscr{A}_1 \cup \mathscr{A}_2.$$
$$\mathscr{A}_1 := \left\{\begin{array}{l}\neg\mathsf{c} \leftarrow \mathsf{a},\\ \neg\mathsf{f} \leftarrow \mathsf{d}\end{array}\right\}.$$
$$\mathscr{A}_2 := \left\{\begin{array}{l}\mathsf{c} \leftarrow \mathsf{b},\\ \mathsf{e} \leftarrow \mathsf{d},\\ \mathsf{f} \leftarrow \mathsf{e}\end{array}\right\}.$$



Let us compare the specificity of the arguments $(\mathscr{A}_1, \mathsf{g}_1)$ and $(\mathscr{A}_2, \mathsf{g}_2)$.

We have $(\mathscr{A}_2, \mathsf{g}_2) \not\lesssim_{\text{CP1}} (\mathscr{A}_1, \mathsf{g}_1)$ because $\{\mathsf{b}, \mathsf{d}\} \subseteq \mathfrak{T}_{\Pi_{13}} = \{\mathsf{a}, \mathsf{b}, \mathsf{d}\}$ is an activation set for $(\mathscr{A}_2, \mathsf{g}_2)$, but not for $(\mathscr{A}_1, \mathsf{g}_1)$.

We have $(\mathscr{A}_1, \mathsf{g}_1) \lesssim_{\text{CP1}} (\mathscr{A}_2, \mathsf{g}_2)$ because, for every activation set $H \subseteq \mathfrak{T}_{\Pi_{13}}$ for $(\mathscr{A}_1, \mathsf{g}_1)$, we have $\{\mathsf{a}, \mathsf{d}\} \subseteq H$; and so H is also an activation set for $(\mathscr{A}_2, \mathsf{g}_2)$.

We again have $(\mathscr{A}_1, \mathsf{g}_1)\ \triangle_{\text{P3}}\ (\mathscr{A}_2, \mathsf{g}_2)$, for the same reason as in Example 12. Thus, the situation for $\lesssim_{\text{P3}}$ is just as in Example 12, and just as "seemingly unintuitive" for exactly the same reason.

We have $(\mathscr{A}_1, \mathsf{g}_1) <_{\text{CP1}} (\mathscr{A}_2, \mathsf{g}_2)$, which is intuitively correct because the conjunction of a more specific and an equivalently specific argument, respectively, should be more specific. Indeed, from the isomorphic sub-specifications in Examples 2 and 3, we know that $(\mathscr{A}_1, \neg\mathsf{c}) <_{\text{CP1}} (\mathscr{A}_2, \mathsf{c})$ and $(\mathscr{A}_1, \neg\mathsf{f}) \approx_{\text{CP1}} (\mathscr{A}_2, \mathsf{f})$, respectively.

All in all, the relation $\lesssim_{\text{P3}}$ fails in this example again, whereas the quasi-ordering $\lesssim_{\text{CP1}}$ works according to human intuition and satisfies monotonicity w.r.t. conjunction.

7.2 Implementation of the preference of the "more precise"

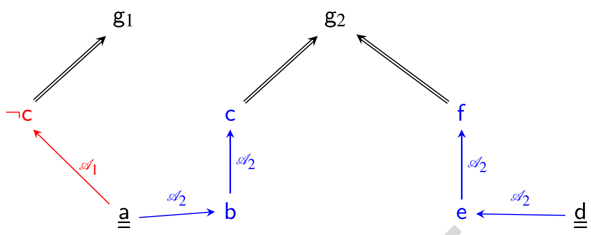
As primary sources of differences in specificity, all previous examples — except Example 4 of Section 3, continued in Example 10 of Section 6.6 — illustrate only the effect of chains of implications. According to our motivating discussion of Section 4.4.5, we should consider also examples where the primary source of differences in specificity is an essentially required condition that is a super-conjunction of the condition triggering another rule. We will do so in the following examples.

As we have already shown in Example 10, both relations $\lesssim_{\text{P3}}$ and $\lesssim_{\text{CP1}}$ produce the intuitive result if the "more precise" super-conjunction is *directly* the condition of a rule. Let us see whether this is also the case if the condition of the rule is *derived* from a super-conjunction.

By removing the second condition literal $\neg\mathsf{f}$ in the strict general rule $\mathsf{g}_1 \Leftarrow \neg\mathsf{c} \wedge \neg\mathsf{f}$ of Example 12, we obtain the following example.

Example 14 (*2nd Variation of Example 12*)

$$\Pi^{\mathrm{F}}_{14} := \{\mathsf{a},\ \mathsf{d}\},$$
$$\Pi^{\mathrm{G}}_{14} := \left\{\begin{array}{l}\mathsf{g}_1 \Leftarrow \neg\mathsf{c},\\ \mathsf{g}_2 \Leftarrow \mathsf{c}\wedge\mathsf{f}\end{array}\right\},$$
$$\Delta_{14} := \mathscr{A}_1 \cup \mathscr{A}_2.$$
$$\mathscr{A}_1 := \{\neg\mathsf{c} \leftarrow \mathsf{a}\}.$$
$$\mathscr{A}_2 := \left\{\begin{array}{l}\mathsf{b} \leftarrow \mathsf{a},\\ \mathsf{c} \leftarrow \mathsf{b},\\ \mathsf{e} \leftarrow \mathsf{d},\\ \mathsf{f} \leftarrow \mathsf{e}\end{array}\right\}.$$



Let us compare the specificity of the arguments $(\mathscr{A}_1, \mathsf{g}_1)$ and $(\mathscr{A}_2, \mathsf{g}_2)$.

We have $(\mathscr{A}_1, \mathsf{g}_1) \not\lesssim_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{g}_2)$ because $\{\mathsf{a}\} \subseteq \mathfrak{T}_{\Pi_{14}} = \{\mathsf{a}, \mathsf{d}\}$ is an activation set for $(\mathscr{A}_1, \mathsf{g}_1)$, but not for $(\mathscr{A}_2, \mathsf{g}_2)$.

We have $(\mathscr{A}_2, \mathsf{g}_2) \lesssim_{\mathrm{CP1}} (\mathscr{A}_1 \mathsf{g}_1)$ because any activation set for $(\mathscr{A}_2, \mathsf{g}_2)$ that is a subset of $\mathfrak{T}_{\Pi_{14}}$ includes a, and so is also an activation set for $(\mathscr{A}_1, \mathsf{g}_1)$.

Considering Theorem 3 as well as the the activation set $\{\mathsf{b}, \mathsf{d}\}$ for $(\mathscr{A}_2, \mathsf{g}_2)$,

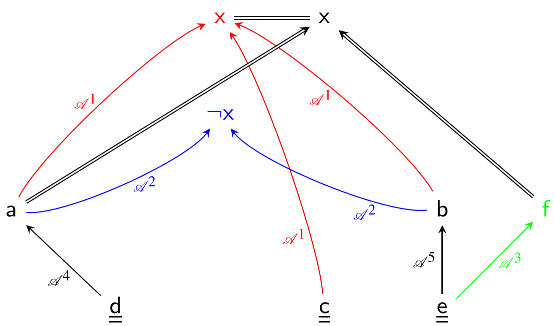
$$\begin{array}{lll}\text{we get} & (\mathscr{A}_1, \mathsf{g}_1) \ \triangle_{\mathrm{P3}} \ (\mathscr{A}_2, \mathsf{g}_2), \\ \text{contrary to} & (\mathscr{A}_1, \mathsf{g}_1) >_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{g}_2).\end{array}$$

Thus, $\lesssim_{\mathrm{CP1}}$ realizes the intuition that the super-conjunction $\mathsf{a} \wedge \mathsf{d}$ — which is essential to derive $\mathsf{c} \wedge \mathsf{f}$ according to $\mathscr{A}_2$ — is more specific than the "less precise" a.

Just like Example 9 of Section 6.6, this example shows again that $\lesssim_{\mathrm{P3}}$ does not properly implement the intuition that — in a model-theoretic approach to specificity — defeasible rules should be considered for their global semantic effect instead of their syntactic fine structure.

Example 15 (*Example 11 from* [27, p. 96])

$$\Pi^{\mathrm{F}}_{15} := \{\mathsf{c},\ \mathsf{d},\ \mathsf{e}\},$$
$$\Pi^{\mathrm{G}}_{15} := \{\mathsf{x} \Leftarrow \mathsf{a}\wedge\mathsf{f}\},$$
$$\Delta_{15} := \mathscr{A}^1 \cup \mathscr{A}^2 \cup \mathscr{A}^3 \cup \mathscr{A}^4 \cup \mathscr{A}^5.$$
$$\mathscr{A}^1 := \{\mathsf{x} \leftarrow \mathsf{a}\wedge\mathsf{b}\wedge\mathsf{c}\}.$$
$$\mathscr{A}^2 := \{\neg\mathsf{x} \leftarrow \mathsf{a}\wedge\mathsf{b}\}.$$
$$\mathscr{A}^3 := \{\mathsf{f} \leftarrow \mathsf{e}\}.$$
$$\mathscr{A}^4 := \{\mathsf{a} \leftarrow \mathsf{d}\}.$$
$$\mathscr{A}^5 := \{\mathsf{b} \leftarrow \mathsf{e}\}.$$



Compare the specificity of the arguments $(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x})$, $(\mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \neg\mathsf{x})$, $(\mathscr{A}^3 \cup \mathscr{A}^4, \mathsf{x})$!

We have $(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x}) <_{\mathrm{CP1}} (\mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \neg\mathsf{x}) \approx_{\mathrm{CP1}} (\mathscr{A}^3 \cup \mathscr{A}^4, \mathsf{x})$, because of $\mathsf{x}, \neg\mathsf{x} \notin \mathfrak{T}_{\Pi_{15}}$, and because any activation set $H \subseteq \mathfrak{T}_{\Pi_{15}} = \{\mathsf{c}, \mathsf{d}, \mathsf{e}\}$ for any of

$(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x})$, $(\mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \neg\mathsf{x})$, $(\mathscr{A}^3 \cup \mathscr{A}^4, \mathsf{x})$ contains $\{\mathsf{d}, \mathsf{e}\}$, which is an activation set only for the latter two.

This matches our intuition well, because the first of these arguments essentially requires the "more precise" $\mathsf{c} \wedge \mathsf{d} \wedge \mathsf{e}$ instead of the less specific $\mathsf{d} \wedge \mathsf{e}$.

We have $(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x})$ $\triangle_{\mathrm{P3}}$ $(\mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \neg\mathsf{x})$ $\triangle_{\mathrm{P3}}$ $(\mathscr{A}^3 \cup \mathscr{A}^4, \mathsf{x})$ $\triangle_{\mathrm{P3}}$ $(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x})$, however. This means that $\lesssim_{\mathrm{P3}}$ cannot compare these counterarguments and cannot help us to pick the more specific argument.

What is most interesting under the computational aspect is that, for realizing

$$(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x}) \not\lesssim_{\mathrm{P3}} (\mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \neg\mathsf{x}),$$

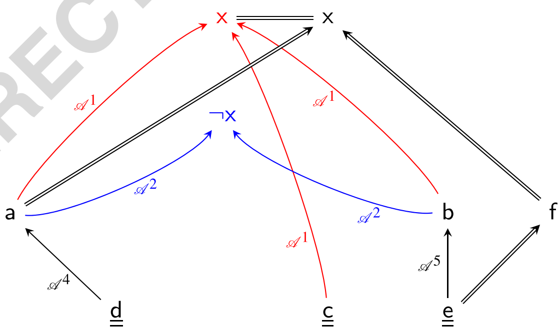
we have to consider the simplified activation set $\{\mathsf{d}, \mathsf{f}\} \subseteq \mathfrak{T}_{\Pi_{15} \cup \Delta_{15}}$ for $(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x})$. This means that here — to realize that $\mathsf{f} \in \mathfrak{T}_{\Pi_{15} \cup \Delta_{15}}$ — we have to take into account the defeasible rule of $\mathscr{A}^3$, which is not part of any of the two arguments under comparison.[30]

Note that such considerations are not required, however, for realizing the properties of $\lesssim_{\mathrm{CP1}}$, because defeasible rules not in the given argument can be completely ignored when calculating the minimal activation sets as subsets of $\mathfrak{T}_{\Pi}$ instead of $\mathfrak{T}_{\Pi \cup \Delta}$. In particular, the complication of *pruning* — as discussed in detail in [27, Section 3.3] — does not have to be considered for the operationalization of $\lesssim_{\mathrm{CP1}}$.

By turning the defeasible rule $\mathsf{f} \leftarrow \mathsf{e}$ of Example 15 into a strict general rule, we obtain the following example.

Example 16 (*Variation of Example 15*)

$$\begin{aligned}
\Pi^{\mathrm{F}}_{16} &:= \{\mathsf{c},\ \mathsf{d},\ \mathsf{e}\},\\
\Pi^{\mathrm{G}}_{16} &:= \left\{\begin{array}{l}\mathsf{x} \Leftarrow \mathsf{a}\wedge\mathsf{f},\\ \mathsf{f} \Leftarrow \mathsf{e}\end{array}\right\},\\
\Delta_{16} &:= \mathscr{A}^1 \cup \mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5,\\
\mathscr{A}^1 &:= \{\mathsf{x} \leftarrow \mathsf{a}\wedge\mathsf{b}\wedge\mathsf{c}\},\\
\mathscr{A}^2 &:= \{\neg\mathsf{x} \leftarrow \mathsf{a}\wedge\mathsf{b}\}.\\
\mathscr{A}^4 &:= \{\mathsf{a} \leftarrow \mathsf{d}\},\\
\mathscr{A}^5 &:= \{\mathsf{b} \leftarrow \mathsf{e}\}.
\end{aligned}$$



Compare the specificity of the arguments $(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x})$, $(\mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \neg\mathsf{x})$, $(\mathscr{A}^4, \mathsf{x})$!

Obviously, $\mathsf{x}, \neg\mathsf{x} \notin \mathfrak{T}_{\Pi_{16}} = \{\mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f}\}$. Moreover, $\{\mathsf{d}\} \subseteq \mathfrak{T}_{\Pi_{16}}$ is an activation set for $(\mathscr{A}^4, \mathsf{x})$ (but not a simplified one!) and, *a fortiori* (by Corollary 5(1)), for $(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x})$, but not for $(\mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \neg\mathsf{x})$. Furthermore, every activation set $H \subseteq \mathfrak{T}_{\Pi_{16}}$ for $(\mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \neg\mathsf{x})$ satisfies $\{\mathsf{d}, \mathsf{e}\} \subseteq H$, which is an activation

[30] Have a look at Fig. 1 in Section 6.1 to see that the effect of f proceeds here only via the set *F*, but not via the usage of the set *H* at the bottom of Fig. 1.

set for $\mathscr{A}^4\mathsf{x}$ and $(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x})$. Finally, every activation set $H \subseteq \mathfrak{T}_{\Pi_{16}}$ for $(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x})$ satisfies $\{\mathsf{d}\} \subseteq H$ which is an activation set for $(\mathscr{A}^4, \mathsf{x})$.

All in all, we have $(\mathscr{A}^4, \mathsf{x}) \approx_{\mathrm{CP1}} (\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x}) >_{\mathrm{CP1}} (\mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \neg\mathsf{x})$.

This is intuitively sound because $(\mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \neg\mathsf{x})$ is activated only by the more specific $\mathsf{d} \wedge \mathsf{e}$, whereas $(\mathscr{A}^4, \mathsf{x})$ is activated also by the "less precise" d.

Moreover, $\mathsf{c} \wedge \mathsf{d} \wedge \mathsf{e}$ is not essentially required for $(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x})$, and so this argument is tantamount to $(\mathscr{A}^4, \mathsf{x})$. The reason for this remarkable effect is not the lack of minimality of the argument $(\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x})$, but our semantic, model-theoretic approach, which simply ignores the fact that the derivation via $\mathscr{A}^1$ requires the more precise activation set. Indeed, we primarily consider consequence, not derivation.

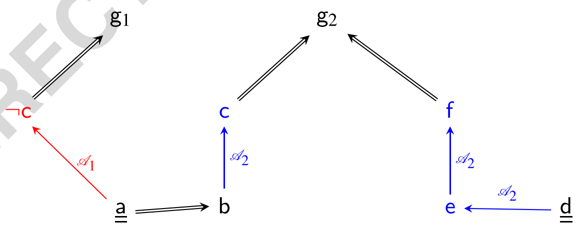
We have $(\mathscr{A}^4, \mathsf{x}) <_{\mathrm{P3}} (\mathscr{A}^1 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \mathsf{x}) \vartriangle_{\mathrm{P3}} (\mathscr{A}^2 \cup \mathscr{A}^4 \cup \mathscr{A}^5, \neg\mathsf{x}) \vartriangle_{\mathrm{P3}} (\mathscr{A}^4, \mathsf{x})$, however. This means that $\lesssim_{\mathrm{P3}}$ fails here completely w.r.t. Poole's intuition, as actually in most non-trivial examples.

7.3 Conflict between the "more concise" and the "more precise"

By removing the second condition literal $\neg\mathsf{f}$ in the strict general rule $\mathsf{g}_1 \Leftarrow \neg\mathsf{c} \wedge \neg\mathsf{f}$ of Example 13, we obtain the following example.

Example 17 (*Variation of Example 13*)

$$\Pi^{\mathrm{F}}_{17} := \{\, \mathsf{a},\ \mathsf{d} \,\},$$
$$\Pi^{\mathrm{G}}_{17} := \left\{ \begin{array}{l} \mathsf{g}_1 \Leftarrow \neg\mathsf{c}, \\ \mathsf{g}_2 \Leftarrow \mathsf{c}\wedge\mathsf{f}, \\ \mathsf{b} \Leftarrow \mathsf{a} \end{array} \right\},$$
$$\Delta_{17} := \mathscr{A}_1 \cup \mathscr{A}_2.$$
$$\mathscr{A}_1 := \{\, \neg\mathsf{c} \leftarrow \mathsf{a} \,\}.$$
$$\mathscr{A}_2 := \{\, \mathsf{c} \leftarrow \mathsf{b},\ \mathsf{e} \leftarrow \mathsf{d},\ \mathsf{f} \leftarrow \mathsf{e} \,\}.$$



$\mathfrak{T}_{\Pi_{17}} = \{\mathsf{a}, \mathsf{b}, \mathsf{d}\}$. Let us compare the specificity of the arguments $(\mathscr{A}_1, \mathsf{g}_1)$ and $(\mathscr{A}_2, \mathsf{g}_2)$.

We have $(\mathscr{A}_1, \mathsf{g}_1) \vartriangle_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{g}_2)$ for the following reasons: $\{\mathsf{a}\} \subseteq \mathfrak{T}_{\Pi_{17}}$ is an activation set for $(\mathscr{A}_1, \mathsf{g}_1)$, but not for $(\mathscr{A}_2, \mathsf{g}_2)$; $\{\mathsf{b}, \mathsf{d}\} \subseteq \mathfrak{T}_{\Pi_{17}}$ is an activation set for $(\mathscr{A}_2, \mathsf{g}_2)$, but not for $(\mathscr{A}_1, \mathsf{g}_1)$.

By Theorem 3 we also get $(\mathscr{A}_1, \mathsf{g}_1) \vartriangle_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{g}_2)$.

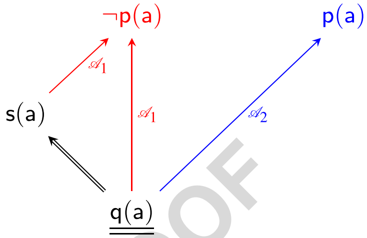
In this example the two intuitive reasons for specificity — super-conjunction (preference of the "more precise") and implication via a strict rule (preference of the "more concise") — are in an irresolvable conflict, which goes well together with the fact that neither $\lesssim_{\mathrm{CP1}}$ nor $\lesssim_{\mathrm{P3}}$ can compare the two arguments.

7.4 Global effect matters more than fine structure

The following example nicely shows that any notion of specificity based only on single defeasible rules (without considering the context of the general strict rules as a whole) cannot be intuitively adequate.

Example 18 (*Example from Page 95 of* [27])

$$\begin{aligned}
\Pi^{\mathrm{F}}_{18} &:= \{\, \mathsf{q}(\mathsf{a}) \,\},\\
\Pi^{\mathrm{G}}_{18} &:= \{\, \mathsf{s}(x) \Leftarrow \mathsf{q}(x) \,\},\\
\Delta_{18} &:= \left\{ \begin{array}{l} \mathsf{p}(x) \leftarrow \mathsf{q}(x),\\ \neg\mathsf{p}(x) \leftarrow \mathsf{q}(x) \wedge \mathsf{s}(x) \end{array} \right\},\\
\mathscr{A}_1 &:= \{\, \neg\mathsf{p}(\mathsf{a}) \leftarrow \mathsf{q}(\mathsf{a}) \wedge \mathsf{s}(\mathsf{a}) \,\},\\
\mathscr{A}_2 &:= \{\, \mathsf{p}(\mathsf{a}) \leftarrow \mathsf{q}(\mathsf{a}) \,\}
\end{aligned}$$



Let us compare the specificity of the arguments $(\mathscr{A}_1, \neg\mathsf{p}(\mathsf{a}))$ and $(\mathscr{A}_2, \mathsf{p}(\mathsf{a}))$.

We have $(\mathscr{A}_1, \neg\mathsf{p}(\mathsf{a})) \approx_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{p}(\mathsf{a}))$, because of $\mathsf{p}(\mathsf{a}), \neg\mathsf{p}(\mathsf{a}) \notin \mathfrak{T}_{\Pi_{18}} = \{\mathsf{q}(\mathsf{a}), \mathsf{s}(\mathsf{a})\}$, and because, for $H \subseteq \mathfrak{T}_{\Pi_{18} \cup \Delta_{18}}$, $i \in \{1, 2\}$, $L_1 := \neg\mathsf{p}(\mathsf{a})$, and $L_2 := \mathsf{p}(\mathsf{a})$, we have the logical equivalence of $H = \{\mathsf{q}(\mathsf{a})\}$ on the one hand, and of H being a minimal simplified activation set for $(\mathscr{A}_i, L_i)$ but not for $(\emptyset, L_i)$, on the other hand.

By Theorem 3, we also get $(\mathscr{A}_1, \neg\mathsf{p}(\mathsf{a})) \approx_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{p}(\mathsf{a}))$.

This makes perfect sense because $\mathsf{q}(\mathsf{a}) \wedge \mathsf{s}(\mathsf{a})$ is not at all strictly "more precise" than $\mathsf{q}(\mathsf{a})$ in the context of Π^{G}_{18}.

Note that nothing is changed here if $\mathsf{s}(x) \Leftarrow \mathsf{q}(x)$ is replaced by setting $\Pi^{\mathrm{G}}_{18} := \{\mathsf{s}(\mathsf{a})\}$. If $\mathsf{s}(x) \Leftarrow \mathsf{q}(x)$ is replaced by setting $\Pi^{\mathrm{G}}_{18} := \emptyset$ and $\Pi^{\mathrm{F}}_{18} := \{\mathsf{q}(\mathsf{a}), \mathsf{s}(\mathsf{a})\}$, however, then we get both $(\mathscr{A}_1, \neg\mathsf{p}(\mathsf{a})) <_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{p}(\mathsf{a}))$ and $(\mathscr{A}_1, \neg\mathsf{p}(\mathsf{a})) <_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{p}(\mathsf{a}))$.

This also speaks for our admission of literals (i.e. unconditional rules) to Π^{G}.[31]

8 Efficiency considerations and the specificity ordering CP2

The specificity relations P1, P2, P3, and CP1[32] share several efficiency features, which we will highlight in this section. Moreover, we will introduce the specificity ordering CP2, a minor variation of CP1 toward more efficiency and intuitive adequacy. Finally, we will discuss further steps toward more efficiency following Herbrand?s Fundamental Theorem.

8.1 A slight gain in efficiency

A straightforward procedure toward deciding the specificity relations $\lesssim_{\mathrm{CP1}}$ and $\lesssim_{\mathrm{P3}}$ between two arguments is to consider all possible activation sets from the literals in the sets $\mathfrak{T}_{\Pi}$ and $\mathfrak{T}_{\Pi \cup \Delta}$, respectively. The effort for computing $\lesssim_{\mathrm{CP1}}$ is lower than that of $\lesssim_{\mathrm{P3}}$ because of $\mathfrak{T}_{\Pi} \subseteq \mathfrak{T}_{\Pi \cup \Delta}$, though not w.r.t. asymptotic complexity: In both cases already the

[31] Cf. Note 1 of Section 2.3.

[32] P1 follows [22] and can be found in this paper in Definition 8 of Section 6.2. P2 follows [26] and can be found in Definition 9 of Section 6.2. P3 respects non-defeasible arguments and can be found in Definition 10 of Section 6.2. CP1 is our transitive relation found in Definition 11 of Section 6.4.

number of possible (simplified) activation sets is exponential in the number of literals in the respective sets $\mathfrak{T}_\Pi$ and $\mathfrak{T}_{\Pi\cup\Delta}$, because each possible subset has to be tested.

8.2 Comparing derivations

To lower the computational complexity, more syntactic criteria for computing specificity were introduced in [27]. These criteria refer to the *derivations* for the given arguments. More precisely, they refer to the *and-trees* of Definition 6 in Section 4.4.1.

8.2.1 No pruning required

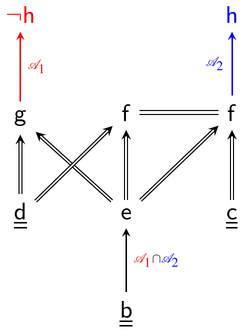
The concept of pruning and-trees is introduced in [27, Definition 12] in this context, because, for the case of $\precsim_{\mathrm{P2}}$, attention cannot be restricted to derivations which make use only of the instances of defeasible rules given in the arguments. The reason for this is that the specificity notions according to [22] and [26] admit literals L in activation sets that cannot be derived solely by strict rules, i.e. $L \in \mathfrak{T}_{\Pi\cup\Delta} \setminus \mathfrak{T}_\Pi$. Since this is not possible with the relation $\precsim_{\mathrm{CP1}}$, this problem vanishes with our corrected version of specificity. This problem and its vanishing are discussed in Example 15 of Section 7.2.

8.2.2 Sets of derivations have to be compared

Yet still, the specificity relation $\precsim_{\mathrm{CP1}}$ inherits several properties from $\precsim_{\mathrm{P3}}$. For instance, the syntactic criteria of their definitions require us in general to compare two *sets* of derivations *element by element*. This is true for both specificity relations:

Example 19 (*Minimal argument with two minimal and-trees/activation sets*)

$$\Pi^{\mathrm{F}}_{19} := \{\,\mathsf{b},\ \mathsf{c},\ \mathsf{d}\,\},$$
$$\Pi^{\mathrm{G}}_{19} := \left\{\begin{array}{l}\mathsf{f} \Leftarrow \mathsf{c}\wedge\mathsf{e},\\ \mathsf{f} \Leftarrow \mathsf{d}\wedge\mathsf{e},\\ \mathsf{g} \Leftarrow \mathsf{d}\wedge\mathsf{e},\end{array}\right\},$$
$$\Delta_{19} := \mathscr{A}_1 \cup \mathscr{A}_2.$$
$$\mathscr{A}_1 := \left\{\begin{array}{l}\neg\mathsf{h} \leftarrow \mathsf{g},\\ \mathsf{e} \leftarrow \mathsf{b}\end{array}\right\}.$$
$$\mathscr{A}_2 := \left\{\begin{array}{l}\mathsf{h} \leftarrow \mathsf{f},\\ \mathsf{e} \leftarrow \mathsf{b}\end{array}\right\}.$$



The argument $(\mathscr{A}_1, \neg\mathsf{h})$ has $\{\mathsf{b}, \mathsf{d}\}$ as the only minimal activation set that is a subset of $\mathfrak{T}_{\Pi_{19}} = \Pi^{\mathrm{F}}_{19}$. $\{\mathsf{b}, \mathsf{d}\}$ is also a minimal activation set for $(\mathscr{A}_2, \mathsf{h})$. On the other hand, $\{\mathsf{b}, \mathsf{c}\}$ is an activation set for $(\mathscr{A}_2, \mathsf{h})$, but not for $(\mathscr{A}_1, \neg\mathsf{h})$. Thus, we get $(\mathscr{A}_1, \neg\mathsf{h}) <_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{h})$.

Because either d or c is in an and-tree of the argument $(\mathscr{A}_2, \mathsf{h})$ (but never both!), a comparison of two fixed and-trees does not suffice.

Moreover note that we have $(\mathscr{A}_1, \neg\mathsf{h}) \vartriangle_{\mathrm{P3}} (\mathscr{A}_2, \mathsf{h})$, because of the simplified activation sets $\{\mathsf{g}\}$ and $\{\mathsf{f}\}$, respectively.

Furthermore note that the only minimal activation set for the minimal argument $(\{\mathsf{e} \leftarrow \mathsf{b}\}, \mathsf{f})$ is $\{\mathsf{b}\}$, which, however, is not a simplified activation set for that argument.

The reason for the complication of an element-by-element comparison of and-trees is that we consider a very general setting of defeasible reasoning in this paper. Indeed, we admit

1. more than one condition literal in rules (conditions containing more than one literal) and
2. non-empty sets of *background knowledge*, i.e. general rules, not only facts.

Typically, only restricted cases are considered: Conditions have always to be singletons in [14], no background knowledge is allowed in [8], and both restrictions are present in [2].

8.2.3 Path criteria?

Before we come to the computation of activations sets via goal-directed derivations in Section 8.3, let us have a closer look here at the path criterion of [27, Section 3.4].

Definition 12 (Path)
For a leaf node N in an and-tree T, we define the *path* in T through N as the empty set if N is the root, and otherwise as the set consisting of the literal labeling N, together with all literals labeling its ancestors except the root node. Let $\mathsf{Paths}(T)$ be the set of all paths in T through all leaf nodes N.

With this notion of paths, the quasi-ordering $\trianglelefteq$ on and-trees can be given as follows:

Definition 13 ([27, Definition 23])
$T_1 \trianglelefteq T_2$ if T_1 and T_2 are two and-trees, and for each $t_2 \in \mathsf{Paths}(T_2)$ there is a path $t_1 \in \mathsf{Paths}(T_1)$ such that $t_1 \subseteq t_2$.

Two and-trees can be compared w.r.t. $\trianglelefteq$ efficiently. This requires the subset comparison of all paths of the two trees, respectively. Hence, the respective complexity is polynomial, at most $\mathrm{O}(n^3)$, where n is the overall number of nodes in the and-trees. This made the relation $\trianglelefteq$ attractive for practical use in the context of [27] compared to the exponential comparison mention in Section 8.1. As stated in the following definition, for a comparison of specificity we have to consider all and-trees, however, and so we still remain with an overall exponential time complexity, which is not better than the one we will describe in Remark 14 of Section 8.3.4.

Definition 14 ([27, Definition 24])
$(\mathcal{A}_1, h_1) \leq (\mathcal{A}_2, h_2)$ if $(\mathcal{A}_1, h_1)$ and $(\mathcal{A}_2, h_2)$ are two arguments in the given specification and for each and-tree T_1 for h_1 there is an and-tree T_2 for h_2 such that $T_1 \trianglelefteq T_2$.

It is shown in [27, Theorem 25] that $\leq$ and $\lesssim_{\mathrm{P2}}$ are equal in special cases, namely if the arguments involved in the comparison correspond to exactly one and-tree. Let us try to adapt this result to our new relation $\lesssim_{\mathrm{CP1}}$, in the sense that we try to establish a mutual subset relation between $\leq$ and $\lesssim_{\mathrm{CP1}}$.

The forward direction is pretty straightforward, but comes with the restriction to be

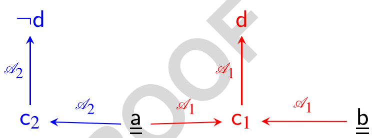
JrnlID 10472_ArtID 9471_Proof#1 - 26/08/2015

expected: From [27, Theorem 25] we get $\leq \subseteq \lesssim_{P2}$. By looking at the empty path, we easily see that $\leq$ satisfies the additional restriction of Definition 10 as compared to Definition 9; so we also get $\leq \subseteq \lesssim_{P3}$. Finally, we can apply Theorem 3 and get the intended $\leq \subseteq \lesssim_{CP1}$, but only with the strong restriction of the condition of Theorem 3. We see no way yet to relax this restriction resulting from phase 3 of Section 6.1.

It is even more unfortunate that the backward direction does not hold at all because of our change in phase 1 of Section 6.1. In particular, as shown in the following example, it does not hold for the special case where it holds for $\lesssim_{P2}$, i.e. in the case that there are no general rules and hence each minimal argument corresponds to exactly one derivation (cf. the proof of Theorem 25 in [27]).

Example 20

$$\Pi^{F}_{20} := \{\mathsf{a},\mathsf{b}\},\quad \Pi^{G}_{20} := \emptyset,$$
$$\Delta_{20} := \mathscr{A}_1 \cup \mathscr{A}_2.$$
$$\mathscr{A}_1 := \{\, \mathsf{c}_1 \leftarrow \mathsf{a}\wedge\mathsf{b},\ \mathsf{d} \leftarrow \mathsf{c}_1 \,\}.$$
$$\mathscr{A}_2 := \{\, \mathsf{c}_2 \leftarrow \mathsf{a},\ \neg\mathsf{d} \leftarrow \mathsf{c}_2 \,\}.$$



We have $(\mathscr{A}_1, \mathsf{d}) \vartriangle_{P3} (\mathscr{A}_2, \neg\mathsf{d})$ and $(\mathscr{A}_1, \mathsf{d}) <_{CP1} (\mathscr{A}_2, \neg\mathsf{d})$.

Both arguments $(\mathscr{A}_1, \mathsf{d})$ and $(\mathscr{A}_2, \neg\mathsf{d})$ correspond to exactly one and-tree, say T_1 and T_2, respectively. All paths in $\mathsf{Paths}(T_1)$ contain c_1, but not c_2, and all paths in $\mathsf{Paths}(T_2)$ contain c_2, but not c_1. Hence, $(\mathscr{A}_1, \mathsf{d}) \leq (\mathscr{A}_2, \neg\mathsf{d})$ does *not* hold.

8.3 Toward a more efficiently realizable notion of Poole-style specificity

Contrary to our small examples in the previous sections, examples of a practically relevant size require notions of specificity that can be decided efficiently.

As we are mainly interested in the more specific arguments, i.e. in the minimal elements of our specificity ordering, we may admit variations of our specificity ordering CP1 that offer better chances for an efficient implementation, but do not relevantly differ w.r.t. these minimal elements.

Therefore, in this section, we will introduce another correction (CP2) of Poole's specificity relation, which offers some advantages for the computation of the respective activation sets, whereas our specificity ordering CP1 offers only the minor advantages over P1, P2, P3 we have already described in Section 8.1 and 8.2.1.

More precisely, our plan for this section is to obtain another quasi-ordering $\lesssim_{CP2}$ by slight modification of our quasi-ordering $\lesssim_{CP1}$, such that the two do not differ in any of our previous examples, and such that $\lesssim_{CP2}$ may mirror our intuition on specificity according to the analysis in Section 4 even more closely in some aspects. Finally, we will try to develop a more efficient procedure for deciding the specificity quasi-ordering $\lesssim_{CP2}$ than those known for any of $\lesssim_{P1}, \lesssim_{P2}, \lesssim_{P3}, \lesssim_{CP1}$.

The crucial step in such a procedure is the computation of activation sets. For a goal-directed, SLD-resolution-like computation of activation sets we cannot keep our restriction to arguments that are ground. For this reason, we now have to modify our notion of a derivation by disallowing the instantiation of variables in our definition of $\mathfrak{T}_{\Pi}$ and $\vdash$ (cf.

Definition 3) as already hinted at in Remark 3 at the end of Section 2.4. As a compensation, we then may add a hat over a set of rules Π, such that $\hat{\Pi}$ denotes the set of all instances of Π.

8.3.1 *Immediate activation sets*

As a first step — since the workaround via path criteria failed in Section 8.2.3 — we now have to find a new notion of an *immediate* activation set such that there are fewer[33] and more easily computable immediate activation sets for a given argument than (non-immediate) activation sets according to Definition 7 of Section 6.1. Our idea here is to avoid SLD-resolution steps that expand a goal clause by *inessential* applications of rules in the sense of the following definition, where we again apply the simple concept of an and-tree given in Definition 6 of Section 4.4.1.

Definition 15 (Inessential Application of an Instance of a Rule)
The application of the instance $L \Leftarrow C$ of a rule in an and-tree is *inessential (in the and-tree)* if there is a node between the root (inclusively) and the application (including the node labeled with L) that is labeled with an element of $\mathfrak{T}_{\hat{\Pi}}$.

As a step toward a more efficiently realizable notion of Poole-style specificity, we will now eliminate those activation sets from our considerations that rely on and-trees with an inessential application of the instance of a defeasible rule.[34]

As a side effect, this step will also eliminate all redundant activation sets that result from what was called "growth of the defeasible parts toward the leaves" in Section 4.4.3. This growth results from inessential application not of defeasible rules, but of general rules only. Contrary to the inessential application of instances of defeasible rules, this elimination of inessential applications of general rules will not change our specificity relation.

The positive effect, however, of cutting off this growth is the following. When the leaves of the defeasible part of an and-tree are included in $\mathfrak{T}_{\hat{\Pi}}$ for the first time in a root-to-leaves traversal, we *immediately* stop and obtain one single immediate activation set, and that's it! The further enumeration of subsumed activation sets is no longer required.

While this reduction of the number of activation sets to one single immediate activation set for each and-tree is most helpful for the computation related to the first argument of the relation $\lesssim_{\mathrm{CP2}}$ when trying to decide it, for the computation related to the second argument it re-introduces the complication we already had in our first sketch of a notion of specificity in Section 4.3.2, as compared to the simplified, second version of this sketch in Section 4.4.4, which was the basis for our first formal definition of activation sets in Definition 7 of Section 6.1.

This complication is only a notational one. It requires the notion of *weakly* immediate activation sets in addition to (non-weakly) immediate ones. This complication does not mean any extra-computation, not even for the second argument in the test for $\lesssim_{\mathrm{CP2}}$: It is just so that the test whether every activation set of the first argument is subsumed by some activation set for the second argument becomes independent from the computation

[33] There are indeed never more (cf. Corollary 7(4)), and typically much less immediate activation sets than activation sets.

[34] The first idea could be to take only activation sets all of whose literals occur in the condition of a rule in $\mathscr{A}$, for the respective argument $(\mathscr{A}, L)$. This idea, however, is too restrictive because also general rules may play a rôle in the defeasible parts of the derivations, cf. Section 4.4.1.

of activation sets. This independence has the advantage that we can optimize it in several directions: First of all, we must omit all rules from Π^{F} and Δ, which play some minor rôles in the computation of non-immediate activation sets (namely Π^{F} for acceptance as an activation set, and the instances of Δ that form the first element of the argument for expansion of activation sets). It is more important, however, that we may also add some forward reasoning from the activation set computed for the first argument in the test for $\precsim_{\mathrm{CP2}}$.

All in all, this means for our operationalization that the computation of activation sets (cf. Definition 7) has to be replaced with the computation of *immediate* activation sets according to the following definition, which also mirrors our isolation of defeasible parts of derivations in Section 4.4.1 more directly than before, namely in the sense that a growth towards the leaves is avoided and the further dissection described in Note 5 of Section 4.4.2 takes place.

It may be helpful for an intuitive understanding of the following definition to have a look at Fig. 1 in Section 6.1: The root tree depicted there is captured in item 2 of the following definition, its sub-trees in item 1.

Definition 16 ([Minimal/Weakly] *Immediate* Activation Set)
Let $\mathscr{A}$ be a set of instances of rules from Δ, and let L be a literal.

H is an *immediate activation set for* $(\mathscr{A}, L)$ if $H \subseteq \mathfrak{T}_{\hat{\Pi}}$ and there is a (possibly empty) set of literals $\mathfrak{L}$, such that both of the following two items hold:

1. For each $L' \in \mathfrak{L}$ there is an and-tree for the derivation of $H \cup \mathscr{A} \cup \hat{\Pi}^{\mathrm{G}} \vdash \{L'\}$ in which
   (a) the root is labeled with L' and generated by an element of $\mathscr{A}$, and
   (b) every literal L'' that labels a non-leaf node or the root satisfies $L'' \notin \mathfrak{T}_{\hat{\Pi}}$, and
   (c) every literal $L'' \notin \mathscr{A}$ that labels a leaf node satisfies $L'' \in \mathfrak{T}_{\hat{\Pi}}$,[35]

   such that the set of literals labeling the leaves of these trees is a subset of $H \cup \mathfrak{T}_{\hat{\Pi}^{\mathrm{G}}} \cup \mathscr{A}$.
2. There is an and-tree for the derivation of $\mathfrak{L} \cup \hat{\Pi} \vdash \{L\}$, such that each literal L'' labeling a node in a path from the root to a leaf labeled with an element from $\mathfrak{L}$ satisfies $L'' \notin \mathfrak{T}_{\hat{\Pi}}$.

H is a *minimal immediate activation set for* $(\mathscr{A}, L)$ if H is an immediate activation set for $(\mathscr{A}, L)$, but no proper subset of H is an immediate activation set for $(\mathscr{A}, L)$.

H is a *weakly immediate activation set for* $(\mathscr{A}, L)$ if $H \subseteq \mathfrak{T}_{\hat{\Pi}}$ and there is an immediate activation set H' for $(\mathscr{A}, L)$ with $H' \subseteq \mathfrak{T}_{H \cup \hat{\Pi}^{\mathrm{G}}}$.

Corollary 7 *Let $\mathscr{A}$ be a set of instances of rules from Δ, and let L be a literal.*

1. *If H is an* [*weakly*] *immediate activation set for $(\mathscr{A}, L)$, then we have $H \subseteq \mathfrak{T}_{\hat{\Pi}}$.*
2. *If H is a minimal immediate activation set for $(\mathscr{A}, L)$, then we have $H \subseteq \mathfrak{T}_{\hat{\Pi}} \setminus (\mathfrak{T}_{\hat{\Pi}^{\mathrm{G}}} \cup \mathscr{A})$.*
3. *Every immediate activation set for $(\mathscr{A}, L)$ is a weakly immediate activation set for $(\mathscr{A}, L)$.*
4. *Every* [*weakly*] *immediate activation set for $(\mathscr{A}, L)$ is an activation set*[36] *for $(\mathscr{A}, L)$.*

[35] Here "literal $L'' \notin \mathscr{A}$" means that L'' is a literal that is not a literal in $\mathscr{A}$, i.e. no conclusion of an unconditional rule from $\mathscr{A}$. Note that, by (a), this excludes any overlap of (b) and (c) (which would result in contradictory requirements): If the root is a leaf, then, by (a), it is labeled with a literal from $\mathscr{A}$.

5. *Every minimal activation set for* $(\mathscr{A}, L)$ *that is an immediate activation set for* $(\mathscr{A}, L)$ *is a minimal immediate activation set for* $(\mathscr{A}, L)$.

Remark 7 (Difference between an Activation Set and an Immediate one)
Regarding the respective specificity orderings, an immediate activation set crucially differs from an activation set as follows: Certain defeasible parts may no longer participate in the derivation, namely those parts that derive a node labeled with an element of $\mathfrak{T}_{\hat{\Pi}}$. This means that those deviations which contain inessential (in the sense of Definition 15) applications of instances of defeasible rules can no longer increase the number of activation sets, i.e. can no longer reduce the specificity of an argument.

We cannot see any reason why the fact that the first element of the argument may also be re-used to re-derive a literal of $\mathfrak{T}_{\hat{\Pi}}$ from $\mathfrak{T}_{\hat{\Pi}}$ should be relevant for the specificity of the argument. Therefore we think that this crucial difference (besides the omission of subsumed activation sets, which effects efficiency only) is in line with common intuition.

Moreover, note that the crucial difference also admits the omission of all defeasible rules whose conclusion is part of the theory $\mathfrak{T}_{\hat{\Pi}}$ when computing immediate activations sets, which does not seem to be possible for (non-immediate) activation sets.

Definition 17 ($\precsim_{\mathrm{CP2}}$: 2$^{\mathrm{nd}}$ Version of our Specificity Relation)
$(\mathscr{A}_1, L_1) \precsim_{\mathrm{CP2}} (\mathscr{A}_2, L_2)$ if $(\mathscr{A}_1, L_1)$ and $(\mathscr{A}_2, L_2)$ are arguments, and we have

1. $L_1 \in \mathfrak{T}_{\hat{\Pi}}$ or
2. $L_2 \notin \mathfrak{T}_{\hat{\Pi}}$ and every $H \subseteq \mathfrak{T}_{\hat{\Pi}}$ that is an [minimal] immediate activation set for $(\mathscr{A}_1, L_1)$ is a weakly immediate activation set for $(\mathscr{A}_2, L_2)$.

To see that nothing essential has changed, compare the following Corollary 8 to Corollary 5 of Section 6.4.

Corollary 8 *If* $(\mathscr{A}_1, L_1)$, $(\mathscr{A}_2, L_2)$ *are arguments with* $\mathscr{A}_1 \subseteq \mathscr{A}_2$, *then any of the following conditions is sufficient for* $(\mathscr{A}_1, L_1) \precsim_{\mathrm{CP2}} (\mathscr{A}_2, L_2)$:

1. $L_1 = L_2$.
2. $L_2 \in \mathfrak{T}_{\hat{\Pi}} \Rightarrow L_1 \in \mathfrak{T}_{\hat{\Pi}}$ *and* $\{L_1\} \cup \hat{\Pi} \vdash \{L_2\}$.
3. $L_1 \in \mathfrak{T}_{\hat{\Pi}}$ *(which is implied by* $\mathscr{A}_1 = \emptyset$ *by Definition 5).*

Remark 8 (Optional Minimality Restriction has No Effect)
Note that the omission of the optional restriction to *minimal* immediate activation sets for $(\mathscr{A}_1, L_1)$ in Definition 17 has no effect on the extension of the defined notion.

Proof Suppose that $L_1, L_2 \notin \mathfrak{T}_{\hat{\Pi}}$, and that H'' is an immediate activation set for $(\mathscr{A}_1, L_1)$. Because the related derivation is finite, we may assume that H'' is finite w.l.o.g. Thus, there is a minimal immediate activation set $H \subseteq H''$ for $(\mathscr{A}_1, L_1)$. If we now assume

[36] Instead of the otherwise required condition that $\mathscr{A}$ *is ground*, we assume here — and will do so in what follows without further mentioning — that the definition of an activation set in Definition 7 of Section 6.1 refers (just as Definition 16 of immediate ones and just as we have changed arguments and derivations in this section) to sets also of *non-ground* instances of defeasible rules in the first element of arguments, but with non-instantiating derivations and theories.

$(\mathcal{A}_1, L_1) \lesssim_{\mathrm{CP2}} (\mathcal{A}_2, L_2)$ with respect to a definition with the optional minimality restriction, then H is a weakly immediate activation set for $(\mathcal{A}_2, L_2)$, i.e. there is an immediate activation set $H' \subseteq \mathfrak{T}_{H \cup \hat{\Pi}^{\mathrm{G}}}$ for $(\mathcal{A}_2, L_2)$, which (because of the monotonicity of our logic) implies $H' \subseteq \mathfrak{T}_{H'' \cup \hat{\Pi}^{\mathrm{G}}}$, i.e. H'' is a weakly immediate activation set for $(\mathcal{A}_2, L_2)$ as well, as was to be shown. □

Remark 9 (Relaxation to a *Weakly* immediate activation set is crucial)
Note that we cannot straightforwardly require H to be a (non-weakly) immediate activation set for $(\mathcal{A}_2, L_2)$ in Definition 17, because otherwise our new relation CP2 would already fail to pass Example 2 of Section 3, in the sense that both arguments there would be incomparable.[37]

Theorem 4 *$\lesssim_{CP2}$ is a quasi-ordering on arguments.*

Proof of Theorem 4
$\lesssim_{\mathrm{CP2}}$ is a reflexive relation on arguments because of Corollary 8.

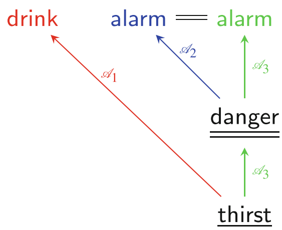
To show transitivity, let us assume $(\mathcal{A}_1, L_1) \lesssim_{\mathrm{CP2}} (\mathcal{A}_2, L_2)$ and $(\mathcal{A}_2, L_2) \lesssim_{\mathrm{CP2}} (\mathcal{A}_3, L_3)$.

According to Definition 17, because of $(\mathcal{A}_1, L_1) \lesssim_{\mathrm{CP2}} (\mathcal{A}_2, L_2)$, we have $L_1 \in \mathfrak{T}_{\hat{\Pi}}$ — and then immediately the desired $(\mathcal{A}_1, L_1) \lesssim_{\mathrm{CP2}} (\mathcal{A}_3, L_3)$ — or we have $L_2 \notin \mathfrak{T}_{\hat{\Pi}}$. The latter case excludes the first option in Definition 17 as a justification for $(\mathcal{A}_2, L_2) \lesssim_{\mathrm{CP2}} (\mathcal{A}_3, L_3)$. Thus, it now suffices to consider the case that $L_i \notin \mathfrak{T}_{\hat{\Pi}}$ for all $i \in \{1, 2, 3\}$.

Suppose that H is an immediate activation set for $(\mathcal{A}_1, L_1)$. It suffices to show that H is a weakly immediate activation set for $(\mathcal{A}_3, L_3)$, i.e. to find an immediate activation set $H'' \subseteq \mathfrak{T}_{H \cup \hat{\Pi}^{\mathrm{G}}}$ for $(\mathcal{A}_3, L_3)$. Because of our supposition, the first step of our original assumption, and the case considered, H is a weakly immediate activation set for $(\mathcal{A}_2, L_2)$, i.e. there is an immediate activation set $H' \subseteq \mathfrak{T}_{H \cup \hat{\Pi}^{\mathrm{G}}}$ for $(\mathcal{A}_2, L_2)$. Then, because of the second step of our original assumption and the case considered, there is an immediate activation set $H'' \subseteq \mathfrak{T}_{H' \cup \hat{\Pi}^{\mathrm{G}}}$ for $(\mathcal{A}_3, L_3)$. Because of the monotonicity of our logic and the closedness of our theories, we now have $H'' \subseteq \mathfrak{T}_{H' \cup \hat{\Pi}^{\mathrm{G}}} \subseteq \mathfrak{T}_{\mathfrak{T}_{H \cup \hat{\Pi}^{\mathrm{G}}} \cup \hat{\Pi}^{\mathrm{G}}} = \mathfrak{T}_{H \cup \hat{\Pi}^{\mathrm{G}}}$, i.e. $H'' \subseteq \mathfrak{T}_{H \cup \hat{\Pi}^{\mathrm{G}}}$, as was to be shown. □

Example 21 ($\lesssim_{\mathrm{CP1}}$ vs. $\lesssim_{\mathrm{CP2}}$)

$$\Pi^{\mathrm{F}}_{21} := \{\, \mathsf{thirst},\ \mathsf{danger} \,\}, \quad \Pi^{\mathrm{G}}_{21} := \emptyset, \quad \Delta_{21} := \mathcal{A}_1 \cup \mathcal{A}_3.$$
$$\mathcal{A}_1 := \{\, \mathsf{drink} \leftarrow \mathsf{thirst} \,\}.$$
$$\mathcal{A}_2 := \{\, \mathsf{alarm} \leftarrow \mathsf{danger} \,\}.$$
$$\mathcal{A}_3 := \mathcal{A}_2 \cup \{\, \mathsf{danger} \leftarrow \mathsf{thirst} \,\}.$$



First note that — because of $\Pi^{\mathrm{G}}_{21} = \emptyset$ — the two notions of an immediate and a weakly immediate activation set coincide here.

[37] See the discussion at the end of Example 21. It might also be interesting to see that the slight modification (via "weakly"), which we need here, occurred already in our first intuitive sketch of a notion of specificity in Section 4.3 — long before the development of the CP2 notion, cf. [34, Section 3.2].

We have $\mathfrak{T}_{\hat{\Pi}_{21}} = \Pi^{\mathrm{F}}_{21}$. Moreover, we have

$$(\mathscr{A}_2, \mathsf{alarm}) <_{\mathrm{CP1}} (\mathscr{A}_3, \mathsf{alarm}) \approx_{\mathrm{CP2}} (\mathscr{A}_2, \mathsf{alarm}) :$$

There is only one minimal activation set for $(\mathscr{A}_2, \mathsf{alarm})$ that is a subset of $\mathfrak{T}_{\hat{\Pi}_{21}}$, namely $\{\mathsf{danger}\}$. It is also a minimal *immediate* activation set for $(\mathscr{A}_2, \mathsf{alarm})$; to see this, take $\mathfrak{L} := \{\mathsf{alarm}\}$ in Definition 16. There are only two minimal activation sets for $(\mathscr{A}_3, \mathsf{alarm})$ that are subsets of $\mathfrak{T}_{\hat{\Pi}_{21}}$, namely $\{\mathsf{danger}\}$ and $\{\mathsf{thirst}\}$, but only the first one is an immediate activation set for $(\mathscr{A}_3, \mathsf{alarm})$. Note that $(\mathscr{A}_2, \mathsf{alarm})$ is *strictly* more specific than $(\mathscr{A}_3, \mathsf{alarm})$ in the sense of $(\mathscr{A}_2, \mathsf{alarm}) \not\gtrsim_{\mathrm{CP1}} (\mathscr{A}_3, \mathsf{alarm})$ by the inessential[38] application of the rule $\mathsf{danger} \leftarrow \mathsf{thirst}$ of $\mathscr{A}_3$, which is not admitted in the definition of *immediate* activation sets and which can be completely ignored in their computation.

Furthermore, we have

$$(\mathscr{A}_1, \mathsf{drink}) <_{\mathrm{CP1}} (\mathscr{A}_3, \mathsf{alarm}) \vartriangle_{\mathrm{CP2}} (\mathscr{A}_1, \mathsf{drink}) :$$

The minimal [immediate] activation set $\{\mathsf{danger}\}$ for $(\mathscr{A}_3, \mathsf{alarm})$ is not an activation set for $(\mathscr{A}_1, \mathsf{drink})$. The only [immediate] activation set for $(\mathscr{A}_1, \mathsf{drink})$ that is a subset of $\mathfrak{T}_{\hat{\Pi}_{21}}$ is $\{\mathsf{thirst}\}$, which is an activation set for $(\mathscr{A}_3, \mathsf{alarm})$, *but not a weakly immediate one*. Note that $(\mathscr{A}_1, \mathsf{drink})$ is no longer more or equivalently specific than $(\mathscr{A}_3, \mathsf{alarm})$ in the sense of $(\mathscr{A}_1, \mathsf{drink}) \not\lesssim_{\mathrm{CP2}} (\mathscr{A}_3, \mathsf{alarm})$, because the inessential application of the rule $\mathsf{danger} \leftarrow \mathsf{thirst}$ of $\mathscr{A}_3$ is not admitted for *immediate* activation sets.

In spite of these minor but noticeable differences, however, nothing has actually changed by stepping from CP1 to CP2, except the positioning of the argument $(\mathscr{A}_3, \mathsf{alarm})$, which is non-minimal as an argument (and therefore practically irrelevant and not even considered in many frameworks, cf. Remark 4 of Section 2.4) and also non-minimal in $\lesssim_{\mathrm{CP1}}$ (and therefore less specific and not really relevant either). What is crucial, however, is that a most specific argument cannot be found in either case. Indeed, we have both

$$\begin{aligned} &(\mathscr{A}_1, \mathsf{drink}) \vartriangle_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{alarm}) \\ \text{and } &(\mathscr{A}_1, \mathsf{drink}) \vartriangle_{\mathrm{CP2}} (\mathscr{A}_2, \mathsf{alarm}). \end{aligned}$$

If we remove danger from Π^{F}_{21}, then $(\mathscr{A}_2, \mathsf{alarm})$ is no argument anymore, but we can embed the specification injectively into the one of Example 3 of Section 3 and get both

$$\begin{aligned} &(\mathscr{A}_1, \mathsf{drink}) \approx_{\mathrm{CP1}} (\mathscr{A}_3, \mathsf{alarm}) \\ \text{and } &(\mathscr{A}_1, \mathsf{drink}) \approx_{\mathrm{CP2}} (\mathscr{A}_3, \mathsf{alarm}), \end{aligned}$$

because the activation set $\{\mathsf{thirst}\}$ now becomes an immediate one also for $(\mathscr{A}_3, \mathsf{alarm})$. Indeed, the application of the rule $\mathsf{danger} \leftarrow \mathsf{thirst}$ is no longer inessential for deriving alarm.

Moreover, if we now add the rule $\mathsf{danger} \Leftarrow \mathsf{thirst}$ to Π^{G}_{21}, resulting in the specification $(\{\mathsf{thirst}\}, \{\mathsf{danger} \Leftarrow \mathsf{thirst}\}, \Delta_{21})$, then the situation is essentially the same as in Example 2 of Section 3, and so we get both $(\mathscr{A}_1, \mathsf{drink}) <_{\mathrm{CP1}} (\mathscr{A}_3, \mathsf{alarm}) \approx_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{alarm})$

$$\text{and } (\mathscr{A}_1, \mathsf{drink}) <_{\mathrm{CP2}} (\mathscr{A}_3, \mathsf{alarm}) \approx_{\mathrm{CP2}} (\mathscr{A}_2, \mathsf{alarm}),$$

because — although the application of the rule $\mathsf{danger} \leftarrow \mathsf{thirst}$ becomes inessential again by $\mathsf{danger} \in \mathfrak{T}_{\hat{\Pi}}$ — $\{\mathsf{thirst}\}$ now becomes a weakly immediate activation set for $(\mathscr{A}_3, \mathsf{alarm})$ and for $(\mathscr{A}_2, \mathsf{alarm})$, though not an immediate one.

[38] This means inessential in the sense of Definition 15.

Corollary 9 ($\lesssim_{\mathrm{CP1}}$ and $\lesssim_{\mathrm{CP2}}$ are incomparable)
There are a specification $(\Pi^{\mathrm{F}}_{21}, \Pi^{\mathrm{G}}_{21}, \Delta_{21})$ *(without any negative literals) and arguments* $(\mathscr{A}_1, L_1)$, $(\mathscr{A}_3, L_3)$, $(\mathscr{A}_2, L_2)$, *such that* $(\mathscr{A}_1, L_1) \lesssim_{CP1} (\mathscr{A}_3, L_3) \lesssim_{CP2} (\mathscr{A}_2, L_2)$

$$\textit{and}\ \ (\mathscr{A}_1, L_1) \not\lesssim_{CP2} (\mathscr{A}_3, L_3) \not\lesssim_{CP1} (\mathscr{A}_2, L_2),$$

i.e. $\lesssim_{CP1} \not\subseteq \lesssim_{CP2} \not\subseteq \lesssim_{CP1}$.

Nevertheless, Example 21 suggests that only some unimportant details make $\lesssim_{\mathrm{CP1}}$ and $\lesssim_{\mathrm{CP2}}$ incomparable to each other, but that the most specific minimal arguments seem to remain most specific and so nothing essential seems to change.

So we may ask ourselves: What changes occur in our previous examples when we switch from CP1 to CP2? Do any of the relations stated for CP1 change for CP2?

The answer to the latter question is: No! We would like to ask the reader to check this carefully.

Example 22 *(continuing Example 18)*
Indeed, the only noticeable change occurs in Example 18, where $\{\mathsf{q(a)}\}$ is a minimal activation set for $(\mathscr{A}_1, \neg\mathsf{p(a)})$, but not an *immediate* activation set. Nevertheless, because $\{\mathsf{q(a)}\}$ is a *weakly* immediate activation set for $(\mathscr{A}_1, \neg\mathsf{p(a)})$, and because the only immediate activation set for $(\mathscr{A}_1, \neg\mathsf{p(a)})$ is $\{\mathsf{q(a)}, \mathsf{s(a)}\}$, which is a weakly immediate activation set for $(\mathscr{A}_2, \mathsf{p(a)})$ (for which $\{\mathsf{q(a)}\}$ is the only immediate one), we have

$$(\mathscr{A}_1, \neg\mathsf{p(a)}) \approx_{\mathrm{CP2}} (\mathscr{A}_2, \mathsf{p(a)}) \text{ as well as } (\mathscr{A}_1, \neg\mathsf{p(a)}) \approx_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{p(a)}).$$

Example 23 (Minimal argument with two minimal *immediate* activation sets)
It is obvious that a minimal argument can easily have two minimal activation sets that are incomparable w.r.t. $\subseteq$. For instance, already in Example 2 of Section 3, the minimal argument $(\mathscr{A}_2, \mathsf{flies(edna)})$ has two minimal [simplified] activation sets, namely $\{\mathsf{bird(edna)}\}$ and $\{\mathsf{emu(edna)}\}$, from which, however, only $\{\mathsf{bird(edna)}\}$ is an *immediate* activation set. In fact, minimal arguments can have more than one minimal *immediate* activation set only if conditions of *general* rules directly contribute to the leaves of the isolated defeasible part as described in Section 4.4.1.[39] This happens in Example 19 of Section 8.2.2 for the minimal argument $(\mathscr{A}_2, \mathsf{h})$: The general rule $\mathsf{f} \Leftarrow \mathsf{c} \wedge \mathsf{e}$ contributes the leaf c to the isolated defeasible part with root h, the inner nodes f and e, and the set of leaves $\{\mathsf{b}, \mathsf{c}\}$, which is one minimal immediate activation set of $(\mathscr{A}_2, \mathsf{h})$. Moreover, the general rule $\mathsf{f} \Leftarrow \mathsf{d} \wedge \mathsf{e}$ contributes the leaf d to the isolated defeasible part with root h, the inner nodes f and e, and the set of leaves $\{\mathsf{b}, \mathsf{d}\}$, which is the other minimal immediate activation set of $(\mathscr{A}_2, \mathsf{h})$, and also the only one for $(\mathscr{A}_1, \neg\mathsf{h})$. Thus, we get both $(\mathscr{A}_1, \neg\mathsf{h}) <_{\mathrm{CP1}} (\mathscr{A}_2, \mathsf{h})$

$$\text{and}\ \ (\mathscr{A}_1, \neg\mathsf{h}) <_{\mathrm{CP2}} (\mathscr{A}_2, \mathsf{h}).$$

[39] Technically, it is possible to enforce a unique immediate activation set for each minimal argument by including the instances also of the *general* rules of the isolated defeasible part into the first element of the arguments. Intuitively, however, this is not reasonable because it leads to unintendedly incomparable arguments.

8.3.2 Special cases with simple activation-set computation

A typical problem in practical application is to classify rules automatically as being facts, general rules, or defeasible rules. We briefly discuss the trivial forms of such a classification now.

The first trivial form of classification is to take all proper rules as defeasible rules. Note that the following lemma (motivated by Example 23 of Section 8.3.1) reduces the task of computing activation sets to the simpler task of computing minimal arguments.

Theorem 5 *Assume that all rules in Π^{G} are just literals (i.e. have empty conditions). Let $(\mathscr{A}, L)$ be a minimal argument. Let $\mathfrak{C}$ be the set of all condition literals of all rules in $\mathscr{A}$. Then $(\mathscr{A}, L)$ has a unique minimal activation set H; and this H is actually a minimal immediate activation set for $(\mathscr{A}, L)$ and equal to $\mathfrak{C} \cap \hat{\Pi}^{\mathrm{F}} \setminus \hat{\Pi}^{\mathrm{G}}$.*

Proof of Theorem 5
Let $(\mathscr{A}, L)$ be a minimal argument.

In case of $L \in \mathfrak{T}_{\hat{\Pi}}$, there is exactly one minimal activation set for $(\mathscr{A}, L)$, namely the empty set, which is an immediate activation set (choose $\mathfrak{L} := \emptyset$ in Definition 16). Moreover, because $(\mathscr{A}, L)$ is a minimal argument, we have $\mathscr{A} = \emptyset$, and then $\mathfrak{C} = \emptyset$. So we get our unique minimal activation set $\emptyset$ indeed in the claimed form of $\mathfrak{C} \cap \hat{\Pi}^{\mathrm{F}} \setminus \hat{\Pi}^{\mathrm{G}} = \emptyset \cap \hat{\Pi}^{\mathrm{F}} \setminus \hat{\Pi}^{\mathrm{G}} = \emptyset$.

It now remains to consider the case of $L \notin \mathfrak{T}_{\hat{\Pi}}$. Because $(\mathscr{A}, L)$ is an argument, there is an and-tree for the derivation of $\hat{\Pi}^{\mathrm{F}} \cup \mathscr{A} \cup \hat{\Pi}^{\mathrm{G}} \vdash \{L\}$. As every and-tree is finite, there is a finite activation set $H' \subseteq \hat{\Pi}^{\mathrm{F}}$ for $(\mathscr{A}, L)$. Then there must be a minimal activation set H for $(\mathscr{A}, L)$ with $H \subseteq H'$. Then we have $H \subseteq \hat{\Pi}^{\mathrm{F}} \setminus \hat{\Pi}^{\mathrm{G}}$. Then there is an and-tree T for the derivation of $H \cup \mathscr{A} \cup \hat{\Pi}^{\mathrm{G}} \vdash \{L\}$ (which is actually unique, but this does not matter here). Let $\mathfrak{D}$ be the set of all conclusions of all rules in $\mathscr{A}$. Let $\mathfrak{D}'$ be the set of all literals in $\mathscr{A}$ (i.e. rules with empty conditions). Then $\mathfrak{D}' \subseteq \mathfrak{D}$. Because $(\mathscr{A}, L)$ is a minimal argument, we know that $\mathfrak{D} \cap \mathfrak{T}_{\hat{\Pi}} = \emptyset$ and that every rule from $\mathscr{A}$ is applied in T. Thus, because of $L \notin \mathfrak{T}_{\hat{\Pi}}$ and because all rules in $\hat{\Pi}$ are just literals, the set of the labels of the leaves of T is exactly $(\mathfrak{C} \cap \mathfrak{T}_{\hat{\Pi}}) \cup \mathfrak{D}'$. Because T is an and-tree for the derivation of $H \cup \mathscr{A} \cup \hat{\Pi}^{\mathrm{G}} \vdash \{L\}$, because $\mathscr{A} \cap \mathfrak{T}_{\hat{\Pi}} \subseteq \mathfrak{D}' \cap \mathfrak{T}_{\hat{\Pi}} \subseteq \mathfrak{D} \cap \mathfrak{T}_{\hat{\Pi}} = \emptyset$, and because all rules in $\hat{\Pi}^{\mathrm{G}}$ are just literals, we have

$$
\begin{aligned}
&(a) & \mathfrak{C} \cap \mathfrak{T}_{\hat{\Pi}} &\subseteq (H \cup \mathscr{A} \cup \hat{\Pi}^{\mathrm{G}}) \cap \mathfrak{T}_{\hat{\Pi}} = H \cup \emptyset \cup \hat{\Pi}^{\mathrm{G}} = H \cup \hat{\Pi}^{\mathrm{G}},\\
&(b) & \mathfrak{T}_{\hat{\Pi}^{\mathrm{G}}} &= \hat{\Pi}^{\mathrm{G}},\\
&(c) & \mathfrak{T}_{\hat{\Pi}} &= \hat{\Pi}^{\mathrm{F}} \cup \hat{\Pi}^{\mathrm{G}}.
\end{aligned}
$$

Because H is a *minimal* activation set for $(\mathscr{A}, L)$, H must be a subset of the leaves of T not in $\mathfrak{D}'$: $H \subseteq \mathfrak{C} \cap \mathfrak{T}_{\hat{\Pi}}$. Because of our previous result of $H \subseteq \hat{\Pi}^{\mathrm{F}} \setminus \hat{\Pi}^{\mathrm{G}}$, we now get $H \subseteq \mathfrak{C} \cap \mathfrak{T}_{\hat{\Pi}} \cap \hat{\Pi}^{\mathrm{F}} \setminus \hat{\Pi}^{\mathrm{G}} \subseteq_{(a)} (H \cup \hat{\Pi}^{\mathrm{G}}) \cap \hat{\Pi}^{\mathrm{F}} \setminus \hat{\Pi}^{\mathrm{G}} = H \cup \emptyset = H$, i.e. $H = \mathfrak{C} \cap \mathfrak{T}_{\hat{\Pi}} \cap \hat{\Pi}^{\mathrm{F}} \setminus \hat{\Pi}^{\mathrm{G}} =_{(c)} \mathfrak{C} \cap (\hat{\Pi}^{\mathrm{F}} \cup \hat{\Pi}^{\mathrm{G}}) \cap \hat{\Pi}^{\mathrm{F}} \setminus \hat{\Pi}^{\mathrm{G}} = \mathfrak{C} \cap \hat{\Pi}^{\mathrm{F}} \setminus \hat{\Pi}^{\mathrm{G}}$. Choosing $\mathfrak{L} := \{L\}$ in item 1 of Definition 16, and a proof tree consisting only of a root in item 2, we see that H is actually an *immediate* activation set for $(\mathscr{A}, L)$; in particular we have $L \notin \mathfrak{T}_{\hat{\Pi}}$ and the property required in the last line of item 1 of Definition 16: $(\mathfrak{C} \cap \mathfrak{T}_{\hat{\Pi}}) \cup \mathfrak{D}' \subseteq_{(a)} H \cup \hat{\Pi}^{\mathrm{G}} \cup \mathscr{A} =_{(b)} H \cup \mathfrak{T}_{\hat{\Pi}^{\mathrm{G}}} \cup \mathscr{A}$. Finally, H is a *minimal* immediate activation set by Corollary 7(5). □

The second trivial form of classification is to take all rules without conditions to be defeasible. It is not a good idea for comparing arguments w.r.t. specificity, however:

Corollary 10 *Assume that $\Pi^{\mathrm{F}} = \emptyset$ and that Π^{G} contains only rules with non-empty conditions. Then we have $\mathfrak{T}_{\hat{\Pi}} = \emptyset$. Moreover, for every argument, there is exactly one* [*immediate*] *activation set H with $H \subseteq \mathfrak{T}_{\hat{\Pi}}$, namely $H = \emptyset$. Furthermore, all arguments are equivalent w.r.t./ $\approx_{CP1}$ and $\approx_{CP2}$.*

Finally, note that the computation of simplified activation sets that are a subset of $\mathfrak{T}_{\hat{\Pi} \cup \hat{\Delta}}$ — as required for P1, P2, P3 instead of CP1, CP2 — is not simplified for the special cases of this section, contrary to the computation of [immediate] activation sets that are subsets of $\mathfrak{T}_{\hat{\Pi}}$.

8.3.3 A step toward operationalization of immediate activation sets

Let us assume that the sets of our predicate and function symbols are enumerable and contain only symbols with finite arities. This assumption does not seem to restrict practical application.

It is straightforward to enumerate for a given input literal — say in a top-down SLD-resolution style — the and-trees of all possible derivations of instances of this input literal, and to interleave this enumeration of and-trees with the enumeration of all ground instances of each and-tree, and finally to enumerate for each ground instance of an and-tree all activation sets for all contained arguments and the ground instance of the input literal labeling the root. Indeed, this is possible because $\mathfrak{T}_{\hat{\Pi}}$ is enumerable (i.e. *semi-decidable*) by our above assumption.

To do the same for all *immediate* activation sets, we have to require the *co-semi-decidability* of $\mathfrak{T}_{\hat{\Pi}}$, because, in general, we cannot output an activation set supposed to be an immediate one before we have established that the literals labeling the ancestors of the nodes of its literals really do *not* occur in $\mathfrak{T}_{\hat{\Pi}}$.

So let us assume the decidability of $\mathfrak{T}_{\hat{\Pi}}$ for the remainder of this section.[40]

It is much harder, however, to enumerate all activation sets in an SLD-like derivation style *directly*, i.e. without storing the intermediate and-trees and their instances. Although *immediate* activation sets offer a crucial advantage for a direct enumeration in principle (because they admit to cut off inessential[41] derivations of literals), the imperative, tail-recursive procedure we will sketch in this section (cf. Fig. 2) still needs further refinement. This procedure enumerates the immediate activation sets *directly*, unless it sometimes outputs the character string `"breach"`, which indicates that some immediate activation sets may be missing.

We present the procedure of Fig. 2 here mainly because we want to concretize the tasks that still remain to be solved for obtaining a Poole-style notion of specificity that admits a sufficiently efficient operationalization, and because our solution of these tasks in Section 8.3.4 may not be the only way to solve them.

Let us assume that *picking* elements from sets satisfies some fairness restriction in the sense that every element will be picked eventually. Moreover, let us assume that we have a procedure to decide $\mathfrak{T}_{\hat{\Pi}}$. Furthermore, let us assume that L is a literal with $L \notin \mathfrak{T}_{\hat{\Pi}}$.

[40] We will relax this restriction in Section 8.3.4.

[41] This means inessential in the sense of Definition 15.

Under these assumptions, the SLD-like procedure immediate-activation-sets(L) of Fig. 2 has the following two properties:

1. If it outputs $(H, (A, I))$ then $I \notin \mathfrak{T}_{\hat{\Pi}}$ is an instance of L, we have $A \neq \emptyset$, and $H \subseteq \mathfrak{T}_{\hat{\Pi}}$ is an immediate activation set for the argument (A, I).
2. If it never outputs "breach", then, for each instance $L\varrho \notin \mathfrak{T}_{\hat{\Pi}}$ with a minimal immediate activation set H' for an argument $(\mathscr{A}, L\varrho)$, it outputs some $(H, (A, I))$ such that there is a substitution μ with $(A\mu, I\mu) = (\mathscr{A}, L\varrho)$ and $H' = H\mu \setminus (\mathfrak{T}_{\hat{\Pi}^{\mathrm{G}}} \cup A\mu)$. As this is similar to what is called a "most general unifier", we may speak of all *maximally general*, immediate activation sets with arguments here.

Remark 10 (Restriction to Ground Conclusions Prevents "breach")
In the special case that the conclusions of all rules of $\Pi^{\mathrm{G}} \cup \Delta$ with non-empty condition are ground, however, the call of the procedure immediate-activation-sets(L) is guaranteed not to output "breach", simply because then only ground literals can enter the set of the program variable O', which are immediately removed again by the line before the tail-recursive call.

Remark 11 (Restriction to Ground Input Literals Does *Not* Prevent "breach")
Note that a restriction to input literals that are ground does not really solve the crucial problem (of which the program variables O, O' have to take care in Fig. 2) that a literal with free variables may be not in $\mathfrak{T}_{\hat{\Pi}}$, whereas some of its instances actually are in $\mathfrak{T}_{\hat{\Pi}}$. The main source of the free variables here are the *extra-variables*, i.e. the free variables that occur in the condition but not in the conclusion of a rule. Such rules with extra-variables and non-ground conclusions, however, are standard in positive-conditional specification, just as in logic programming. A single extra-variable in an arbitrary rule of $\Pi^{\mathrm{G}} \cup \Delta$ can force SLD-resolution to work on non-ground goals even for a ground input literal.

Some examples may be more appropriate here than a proof of the soundness of the procedure of Fig. 2 (that enumerates a maximally general, immediate activation set for each minimal immediate activation set unless it sometimes indicates "breach"), because we see the procedure only as a step in a further development toward a tractability that is sufficient in practice. Therefore, we will give some examples here on how the procedure

$$\text{immediate-activation-sets}(L)$$

works for certain literals $L \notin \mathfrak{T}_{\hat{\Pi}}$, namely by

listing all calls of the auxiliary procedure immediate-activation-sets-helper.

Example 24 *(continuing Example 3 of Section 3)*
Let us consider Example 3 of Section 3. A call of immediate-activation-sets(flies(y)) results in a call of immediate-activation-sets-helper with the argument quintuple

$$(\{(\mathsf{flies}(y), 2)\}, \emptyset, \emptyset, \emptyset, \mathsf{flies}(y)),$$

where the only rule whose conclusion is unifiable with the only goal literal is a defeasible one, namely flies(x) $\leftarrow$ bird(x) from Δ_3. We can take ξ and σ as the identity and $\{x \mapsto y\}$, respectively. The program variable B' will be set to 1, and the tail-recursive call will have the argument tuple

$$(\{(\mathsf{bird}(y), 1)\}, \{\mathsf{flies}(y)\}, \emptyset, \{\mathsf{flies}(y) \leftarrow \mathsf{bird}(y)\}, \mathsf{flies}(y)).$$

procedure immediate-activation-sets(L):
(* L must be a literal *)
  if $L \notin \mathfrak{T}_{\hat{\Pi}}$ then (call immediate-activation-sets-helper$(\{(L,2)\},\emptyset,\emptyset,\emptyset,L)$).

procedure immediate-activation-sets-helper(T,O,H,A,I):
(* T is the current goal. T must be a set of pairs (L,B) of a literal $L \notin \mathfrak{T}_{\hat{\Pi}}$ and
  a bit $B \in \{1,2\}$ referring to the two items of Definition 16,
  such that $B=1$ indicates that L labels a defeasible part *)
(* O is a set of literals that indicate that our algorithm may have missed
  to enumerate a most general immediate activation set in case of $O \cap \mathfrak{T}_{\hat{\Pi}} \neq \emptyset$
  because the and-tree has already been properly expanded at their nodes
  (which occur in a defeasible part!) *)
(* H is an accumulator for the immediate activation set,
  H must always be a set of literals $L \in \mathfrak{T}_{\hat{\Pi}}$ from the fringes of defeasible parts *)
(* A is an accumulator for the first element of the argument *)
(* I is the possibly instantiated input literal and second element of the argument *)
  if $T=\emptyset$ then (output "`H is immediate activation set for` (A,I)" and exit);
  pick some (L,B) from T; $T := T \setminus \{(L,B)\}$;
  for each rule $(L' \Leftarrow L''_1 \wedge \ldots \wedge L''_n) \in \Pi \cup \Delta$ do
  for some ξ that maps all variables in $L' \Leftarrow L''_1 \wedge \ldots \wedge L''_n$ to fresh variables do
  if L and $L'\xi$ have the most general unifier σ then [
    $I' := I\sigma$; if $I' \in \mathfrak{T}_{\hat{\Pi}}$ then (output "`Instance` $I' \in \mathfrak{T}_{\hat{\Pi}}$" and exit);
    $O' := O\sigma$; if $O' \cap \mathfrak{T}_{\hat{\Pi}} \neq \emptyset$ then (output "`breach`" and exit);
    $T' := \{\, (L'''\sigma, B''') \mid (L''', B''') \in T \wedge L'''\sigma \notin \mathfrak{T}_{\hat{\Pi}} \,\}$;
    $H' := H\sigma \cup \{\, L'''\sigma \mid (L''', 1) \in T \wedge L'''\sigma \in \mathfrak{T}_{\hat{\Pi}} \,\}$;
    $A' := A\sigma$;
    if $L\sigma \in \mathfrak{T}_{\hat{\Pi}}$ then (if $B=1$ then $(H' := H' \cup \{L\sigma\})$)
    else (
      $B' := B$;
      if $(L' \Leftarrow L''_1 \wedge \ldots \wedge L''_n) \notin \Pi$ then (
        (* The applied rule is necessarily a defeasible one! *)
        $A' := A' \cup \{(L' \Leftarrow L''_1 \wedge \ldots \wedge L''_n)\xi\sigma\}$;
        $B' := 1$);
      $T' := T' \cup \{\, (L''_i\xi\sigma, B') \mid i \in \{1,\ldots,n\} \wedge L''_i\xi\sigma \notin \mathfrak{T}_{\hat{\Pi}} \,\}$;
      if $B'=1 \wedge n \geq 1$ then (
        (* $B'=1$ means that we are in a defeasible part now,
          and so we have to accumulate our activation set! *)
        (* $n \geq 1$ means that we have to expand the and-tree *properly*
          under the crucial assumption that $L\sigma \notin \mathfrak{T}_{\hat{\Pi}}$. *)
        $H' := H' \cup \{\, L''_i\xi\sigma \mid i \in \{1,\ldots,n\} \wedge L''_i\xi\sigma \in \mathfrak{T}_{\hat{\Pi}} \,\}$;
        $O' := O' \cup \{L\sigma\}$));
    $O' := \{\, L''' \in O' \mid L'''$ is not ground $\}$;
    call immediate-activation-sets-helper(T', O', H', A', I')].

Fig. 2 Sketch of immediate-activation-sets and immediate-activation-sets-helper

Again the only rule whose conclusion is unifiable with the only goal literal is a defeasible one, namely $\mathsf{bird}(x) \leftarrow \mathsf{emu}(x)$ from Δ_3. We can again take ξ and σ as the identity and $\{x \mapsto y\}$, respectively. The program variable B' will be set to 1, and the tail-recursive call will have the argument tuple

$$(\{(\mathsf{emu}(y), 1)\}, \{\mathsf{flies}(y), \mathsf{bird}(y)\}, \emptyset, \{\mathsf{flies}(y) \leftarrow \mathsf{bird}(y), \mathsf{bird}(y) \leftarrow \mathsf{emu}(y)\}, \mathsf{flies}(y)).$$

procedure ground-immediate-activation-sets-helper(T,H,A):
(* T is the current goal. T must be a set of pairs (L,B) of a literal $L \notin \mathfrak{T}_{\Pi_{\rm g}}$ and
  a bit $B \in \{1,2\}$ referring to the two items of Definition 16,
  such that $B = 1$ indicates that L labels a defeasible part *)
(* H is an accumulator for the immediate activation set. H must always be
  a set of literals $L \in \mathfrak{T}_{\Pi_{\rm g}} \setminus \mathfrak{T}_{\Pi^{\rm G}_{\rm g}}$ from the fringes of defeasible parts. *)
(* A is an accumulator for the first element of the argument with $A \cap \mathfrak{T}_{\Pi_{\rm g}} = \emptyset$. *)
(* note that the input literal I is invariant now; no input, no output *)
  if $T = \emptyset$ then (output (H,A) and exit);
  pick some (L,B) from T; $T := T \setminus \{(L,B)\}$;
  (* We do not have to test rules from $\Pi^{\rm F}_{\rm g}$ because of $L \notin \mathfrak{T}_{\Pi_{\rm g}}$. *)
  for each rule $(L' \Leftarrow L''_1 \wedge \ldots \wedge L''_n) \in \Pi^{\rm G}_{\rm g} \cup \Delta_{\rm g}$ do
  if $L = L'$ then [
    $H' := H$; $A' := A$; $B' := B$;
    if $(L' \Leftarrow L''_1 \wedge \ldots \wedge L''_n) \notin \Pi^{\rm G}_{\rm g}$ then (
      (* The applied rule is now necessarily a defeasible one. *)
      $A' := A' \cup \{(L' \Leftarrow L''_1 \wedge \ldots \wedge L''_n)\}$;
      $B' := 1$);
    $T' := T \cup \{\ (L''_i, B') \mid i \in \{1,\ldots,n\} \wedge L''_i \notin \mathfrak{T}_{\Pi_{\rm g}}\ \}$;
    if $B' = 1$ then (
      (* $B' = 1$ means that we are in a defeasible part now,
        and so we have to accumulate our activation set! *)
      $H' := H' \cup \{\ L''_i \mid i \in \{1,\ldots,n\} \wedge L''_i \in \mathfrak{T}_{\Pi_{\rm g}} \setminus \mathfrak{T}_{\Pi^{\rm G}_{\rm g}}\ \}$);
    call ground-immediate-activation-sets-helper(T',H',A')].

Fig. 3 Sketch of procedure ground-immediate-activation-sets-helper

Now the only rule whose conclusion is unifiable with the only goal literal is a fact, namely emu(edna) from $\Pi^{\rm F}_3$. We can take ξ and σ as the identity and $\{y \mapsto$ edna$\}$, respectively. The program variable B' will be set to 1, and the tail-recursive call will have the argument tuple

$$(\emptyset, \emptyset, \{\mathsf{emu(edna)}\}, \{\mathsf{flies(edna)} \leftarrow \mathsf{bird(edna)}, \mathsf{bird(edna)} \leftarrow \mathsf{emu(edna)}\}, \mathsf{flies(edna)}).$$

This call immediately terminates by outputting the immediate activation set {emu(edna)} for the argument ({flies(edna) ← bird(edna), bird(edna) ← emu(edna)}, flies(edna)). As all calls are terminated now and there was no output `"breach"`, this means that we have enumerated all immediate activation sets for all instances of the input literal.

Example 25 *(continuing Example 2 of Section 3)*
Let us now come to Example 2 of Section 3. We start with the same input as for Example 24 above, and there is no change up to the call with argument tuple

$$(\{(\mathsf{bird}(y), 1)\}, \{\mathsf{flies}(y)\}, \emptyset, \{\mathsf{flies}(y) \leftarrow \mathsf{bird}(y)\}, \mathsf{flies}(y)),$$

and the only difference before the next call is that the applied rule is a strict one and is not recorded in the program variable A'. Thus, we get a call with the argument tuple

$$(\{(\mathsf{emu}(y), 1)\}, \{\mathsf{flies}(y), \mathsf{bird}(y)\}, \emptyset, \{\mathsf{flies}(y) \leftarrow \mathsf{bird}(y)\}, \mathsf{flies}(y)).$$

There is still no essential change, except that the test for `"breach"` becomes positive: We again have $O\sigma$ = {flies(edna), bird(edna)}, but now we have bird(edna) $\in \mathfrak{T}_{\hat{\Pi}}$, and our procedure outputs `"breach"`. Indeed, it missed to enumerate the immediate activation set $\{\mathsf{bird(edna)}\}$ for the argument $(\{\mathsf{flies(edna)} \leftarrow \mathsf{bird(edna)}\}, \mathsf{flies(edna)})$, simply because the instantiation came too late to stop us from proper expansion of the and-tree.

Remark 12 (Closer Matching of Activation Sets to SLD-Resolution Results in Inappropriate Semantics)
The obvious idea to avoid the possibility that the procedure of Fig. 2 may output `"breach"` and miss some maximally general, immediate activation sets is the following.

Just like we obtained CP2 from CP1, it is possible to obtain a notion CP3 from CP2 by a minor modification of immediate activation sets, resulting in, say, *SLD activation sets*, such that the SLD-like computation of Fig. 2 enumerates all maximally general, SLD activation sets.

We do not see a chance to satisfy the crucial requirement of such a modification, however, namely that it does not affect any of our previous examples. If we look at the application of the procedure of Fig. 2 to the specification of Example 2 as described in Example 25, then we see that all SLD activation sets remaining in Example 2 could be $\{\mathsf{emu(edna)}\}$, such that the arguments $(\mathscr{A}_1, \neg\mathsf{flies(edna)})$ and $(\mathscr{A}_2, \mathsf{flies(edna)})$ would become equivalently specific w.r.t. the specification of Example 2, which seems to be absurd.

8.3.4 A specificity relation based on given and-trees

We see no straightforward procedure to decide $\lesssim_{\mathrm{CP2}}$. Even worse, we see neither a procedure to semi-decide it, nor a procedure to co-semi-decide it. A positive answer can be given if the procedure of Fig. 2 terminates for the first argument of $\lesssim_{\mathrm{CP2}}$ without outputting `"breach"`. A negative answer can be given if, for an immediate activation set enumerated for the first argument, the derivation for testing the property of being a weakly immediate activation set for the second argument terminates with failure. In general, even if we assume $\mathfrak{T}_{\hat{\Pi}}$ to be decidable, none of these terminations is guaranteed.[42]

In such a situation it is clearly appropriate to relax our requirement of a *model-theoretic* specificity relation a bit. So we replace the fancied decision procedure for $\mathfrak{T}_{\hat{\Pi}}$ with the test whether the literal has a derivation from those instances of Π which can be found in some and-tree occurring in *a finite set of and-trees fixed in advance*. For the solution we are aiming at, it is crucial that this given finite set of and-trees cannot be further extended during related specificity considerations. A good candidate may be the set of those and-trees that our derivation procedure has been able to construct within a certain time limit. Then we can replace each of the three elements of our specification $(\Pi^{\mathrm{F}}, \Pi^{\mathrm{G}}, \Delta)$ with the sets of those instances of their elements that are actually applied in our finite set of and-trees, resulting in the new specification $(\Pi^{\mathrm{F}}_{\mathrm{g}}, \Pi^{\mathrm{G}}_{\mathrm{g}}, \Delta_{\mathrm{g}})$. The further considerations must use these three finite sets without any further instantiation. This means that their rules are to be considered to be ground and this is what the lower index "g" stands for.

We again abbreviate $\Pi_{\mathrm{g}} := \Pi^{\mathrm{F}}_{\mathrm{g}} \cup \Pi^{\mathrm{G}}_{\mathrm{g}}$, and also replace the typically undecidable set $\mathfrak{T}_{\hat{\Pi}}$ with finite set $\mathfrak{T}_{\Pi_{\mathrm{g}}}$.

Note that hardly anything has changed for our set of defeasible rules, because arguments work anyway with instances that are ground, or are at least treated as if they were ground (cf. Remark 3 in Section 2.4), and we can hardly consider an argument that is not contained in some and-tree we have constructed in advance.

[42] Both of these terminations can be guaranteed, however, under most restrictive conditions, such as the one that the conclusions of every rule from $\Pi^{\mathrm{G}} \cup \Delta$ with a non-empty condition are ground (cf. Remark 10).

There is a major change, however, for the set Π of strict rules. The situation here is similar to an expansion w.r.t. a *champ fini* in Herbrand?s Fundamental Theorem,[43] and we have reason to hope that the effect of this change can be neglected in practice, provided that a sufficient number of the proper instances is considered. Note that, for first-order logic, the depth limit n for terms required for Herbrand's Property C to establish a sentential tautology (i.e. the natural number n for the *champ fini* of order n) is not computable in the sense of a *total* recursive function. Even if we knew the smallest such n, however, the number of terms of depth smaller than n would still be too high for practical feasibility in general. This means that it is crucial to choose the instances of our rules in a clever way, say from the successful proofs delivered by a theorem-proving system within a sufficient time limit.

Remark 13 (Specificity Relation on Arguments Extended with an And-Tree)
A straightforward idea to improve tractability is to attach an and-tree to each argument and to compute a unique (cf., however, Example 23 in Section 8.3.1) immediate activation set for each argument as follows: Starting from the root, we traverse the tree, remembering whether we have passed an application of the instance of a defeasible rule, and stop traversing at the first node labeled with an element of the finite set $\mathfrak{T}_{\Pi_g}$, outputting its literal as part of the single *tree-immediate activation set*, provided that we have passed an application of the instance of a defeasible rule.

The problem we see here, however, is that such a fixed and-tree does not make much sense for the second argument of our relation $\precsim_{CP2}$, simply because we should not let an inappropriately chosen and-tree for the second argument produce a failure of the property of being more specific, cf. Example 19 of Section 8.2.2. This means that we need an existential quantification over the and-tree of the second argument. If we were able to find a way to handle this quantification, the same technique would probably admit us to handle a universal quantification over the and-tree of the first argument, which brings us back to our original relation $\precsim_{CP2}$ on arguments without and-trees. So this restriction to concrete and-trees does not seem to help. We will now show that we do not need it either.

With the modifications described above, let us now come back to our procedure of Fig. 2. As noted before (cf. Remark 10), there cannot be any output of `"breach"` anymore, because our new sets of general strict and defeasible rules, i.e. the sets Π_g^G and Δ_g, are now ground by definition. After the resulting simplifications, the procedure immediate-activation-sets-helper now may be replaced with the procedure ground-immediate-activation-sets-helper sketched in Fig. 3.

To ensure termination of ground-immediate-activation-sets-helper we additionally have to store the current path of the and-tree and exit without further output if we encounter a literal for a second time on the same path.

Regarding time complexity of the procedure of Fig. 3 extended with the storage of the current path of the and-tree for ensuring termination mentioned above, only the following preliminary remarks apply in this early state of development.

Remark 14 (Considerations on Complexity)
From practical experience, complexity is not relevant yet: Our straightforward PROLOG (cf. e.g. [6]) implementation of this procedure (which prefers simplicity of coding over efficiency) computes, compares, and sorts — without any noticeable delay in the

[43] Cf. [16, 30–32, 36, 37].

answer — all minimal immediate activation sets for all minimal arguments for all literals of $\mathfrak{T}_{\Pi_g \cup \Delta_g} \setminus \mathfrak{T}_{\Pi_g}$, for a specification $(\Pi_g^F, \Pi_g^G, \Delta_g)$ of all instances required for a superset of all examples in this paper.

Regarding the theoretical worst case, which will hardly ever occur in practice, the following first estimate may be not completely irrelevant. Let n be the number of different literals in all conclusions of all rules of $\Pi_g \cup \Delta_g$. With our mentioned mechanism for ensuring termination, it is obvious that n limits the maximal depth of the SLD-like search tree. Let m be the maximal number of all condition literals of all rules with an identical conclusion. It is obvious that m limits the maximal number of children of any node in the SLD-like search tree, cumulated over the whole run. This means that the maximal size of the cumulated search tree is $m^{n-1} - 1$, i.e. $O(m^n)$. Luckily, this Landau-O limits also the size of the theory $\mathfrak{T}_{\Pi_g}$ (which we pre-compute in our PROLOG implementation) and all other efforts at each node, such as indexing our rules for obtaining a constant effort at each node. Therefore, the whole algorithm is $O(m^n)$.

Remark 15 (Completeness of the Procedure)
Our procedure is complete in the sense that we can compute the finite set of all minimal[44] immediate activation sets of all minimal arguments for a given input literal w.r.t. our ground specification $(\Pi_g^F, \Pi_g^G, \Delta_g)$. All what is left for deciding $\precsim_{CP2}$ is to check whether each of the computed immediate activation sets whose defeasible rules are part of the first argument is a weakly immediate activation set for the second argument. This is straightforward, although it is not clear yet which implementation will be optimal.

We should not forget, however, that the specification $(\Pi_g^F, \Pi_g^G, \Delta_g)$ is only a reasonably constructed sub-specification of our original specification (Π^F, Π^G, Δ), which actually stands for $(\hat{\Pi}^F, \hat{\Pi}^G, \hat{\Delta})$. Practical tests have to show whether such an omission of infinitely many instances can be viable without deteriorating our specificity ordering. Theoretically, such a viability can only be guaranteed for the special case that the number of instances of the rules of the specification is finite (up to renaming of variables).

9 Conclusion

9.1 Summary

We would need further discussions on our surprising new findings w.r.t. Poole's specificity relation, in particular its lack of transitivity. After all, defeasible reasoning with Poole's notion of specificity is being applied now for over a quarter of a century, and it was not to be expected that our investigations could shake an element of the field to the very foundations.

One remedy for the discovered lack of transitivity of $\precsim_{P3}$ could be to consider the transitive closure of the non-transitive relation $\precsim_{P3}$. This could be an advantage compared to $\precsim_{CP1}$ only under the condition that the transitive closure of $\precsim_{P3}$ is a subset of $\precsim_{CP1}$, i.e. only under one of the conditions of Theorem 3. Moreover, this transitive closure still has

[44] *Minimal* immediate activation sets are obtained after completion of the procedure of Fig. 3 simply as follows: For each minimal argument $(\mathscr{A}, L)$, we remove all proper supersets among the immediate activation sets. Note that we do not have to filter the immediate activation sets by removing all elements of $\mathscr{A}$, simply because, as subsets of $\mathfrak{T}_{\Pi_g}$, they are disjoint from the literals in $\mathscr{A}$ (i.e. the rules in $\mathscr{A}$ with empty conditions).

all the the intuitive shortcomings made obvious for $\lesssim_{\mathrm{P3}}$ in Section 7. Furthermore, we do not see how this transitive closure could be decided efficiently. Finally, the transitive closure lacks a direct intuitive motivation, and after the first extension step from $\lesssim_{\mathrm{P3}}$ to its transitive closure, we had better take the second extension step to the more intuitive $\lesssim_{\mathrm{CP1}}$ immediately.

Contrary to the transitive closure of $\lesssim_{\mathrm{P3}}$, our novel relations $\lesssim_{\mathrm{CP1}}$ and $\lesssim_{\mathrm{CP2}}$ also solve the problem of non-monotonicity of specificity w.r.t. conjunction (cf. Section 7.1), which was already realized as a problem of $\lesssim_{\mathrm{P1}}$ by [22] (cf. our Example 12 in Section 7.1).

The present means to decide our novel specificity relation $\lesssim_{\mathrm{CP1}}$, however, show several improvements[45] and a few setbacks[46] compared to the known ones for Poole's relation. Further work is needed to improve efficiency.

By a minor restriction of activation sets, resulting in *immediate* activation sets, we have come in Section 8.3 to the quasi-ordering $\lesssim_{\mathrm{CP2}}$, which does not show any difference compared to $\lesssim_{\mathrm{CP1}}$ in any of our examples except Example 21, which was constructed to show the difference. The new specificity ordering $\lesssim_{\mathrm{CP2}}$ has advantages w.r.t. intuition and efficiency. The latter advantage, however, requires decidability of $\mathfrak{T}_{\hat{\Pi}}$ (in addition to the always given semi-decidability).

To concretize the problems of computing activation sets by SLD-resolution, in Section 8.3.3 we have sketched a procedure that indicates `"breach"` if it may have missed to output some of the most general immediate activation sets. Then, in Section 8.3.4, we have shown how to obtain decidability of $\mathfrak{T}_{\hat{\Pi}}$ by restriction to a finite set of instances that are then treated as if they were ground. We hope that we can find a procedure for generating the finite set of rule instances such that the effect of this restriction can be neglected in practice. Without such a restriction, however, we do not know how to decide any of the relations $\lesssim_{\mathrm{P1}}$, $\lesssim_{\mathrm{P2}}$, $\lesssim_{\mathrm{P3}}$, $\lesssim_{\mathrm{CP1}}$, $\lesssim_{\mathrm{CP2}}$ in general.

9.2 Application contexts

We can apply the specificity relations to question answering, as attempted in the RatioLog project [10]. Question answering systems such as LogAnswer [9] usually determine several possible answer candidates for a given query. For each candidate, a possibly defeasible derivation of the answer is available. The best answer candidate has to be chosen. One idea among others is to prefer more specific answers. Thus, specificity is incorporated as a mechanism of rationality here.

An important part of the application context for specificity orderings consists of numerous frameworks for argumentation in logic. The overall process usually includes a dialectical process used for answering queries. Different arguments are pro or contra a certain answer. By means of an attack relation, conflicts between contradicting arguments can be determined in abstract argumentation frameworks, such as the ones of [7, 23], and [21]. A concrete specificity ordering or a similar relation helps then to decide among conflicting arguments.

The ASPIC+ framework [21] combines an (abstract) argumentation system with a concrete knowledge base, which may contain strict and defeasible rules. In this context, an argument can be attacked on a conclusion of a defeasible inference, on a defeasible inference step itself, or on an ordinary premise. Nonetheless, also ASPIC+ is not a concrete system

[45] See Section 8.1, 8.2.1, 8.3.2, 8.3.3, and 8.3.4 for the improvements.

[46] See Section 8.2.3 and 8.3.3 for the setbacks.

but a framework for specifying systems. The choice of the logic is left open in ASPIC+. Thus, on the basis of the different rule types, the attack or conflict relation may be defined, e.g. by means of one of our specificity orderings.

As the discussion in this paper demonstrates, however, it is not that easy to find an effective concrete specificity relation. One of the main problems is that such relations are often computationally highly complex, such as it is the case in [17].

9.3 More conservative instead of more specific?

Note that we have to distinguish between orderings for comparing conflicting arguments w.r.t. specificity and orderings for comparing arguments w.r.t. a form of subsumption, such as the quasi-ordering of being "more conservative" found in [3, Definition 3.3, p. 206], [4, Definition 6, p. 50]. There, roughly speaking, an argument $(\mathscr{A}_1, L_1)$ is *more conservative* than an argument $(\mathscr{A}_2, L_2)$ if $\mathscr{A}_1 \subseteq \mathscr{A}_2$ and $\{L_2\} \vdash \{L_1\}$. So if our opponent accepts the argument $(\mathscr{A}_2, L_2)$, then he also has to accept our more conservative argument $(\mathscr{A}_1, L_1)$, because we need less presuppositions and our result follows from our opponent's result. In many practical situations, however, the *less* conservative argument will be preferred. For instance, if we ask a question-answering system (such as LogAnswer [9]) for the mother of Pierre Fermat, then — as an answer — we prefer the less conservative argument

$$(\mathscr{A}, \mathsf{Mother}(\text{Claire de Long, Pierre Fermat})) \quad \text{to}$$
$$(\mathscr{A}, \exists x.\mathsf{Mother}(x, \text{Pierre Fermat})).$$

Moreover, the arguments

$$(\mathscr{A}, \mathsf{Mother}(\text{Françoise Cazeneuve, Pierre Fermat})) \quad \text{and}$$
$$(\mathscr{A}, \mathsf{Mother}(\text{Claire de Long, Pierre Fermat})),$$

are incomparable in the "more conservative"-quasi-ordering.[47]

Even worse, for a non-trivial derivability relation, i.e. in a non-contradictory theory, the quasi-ordering of being "more conservative" cannot compare arguments with contradictory results L, $\neg L$ by definition.

Moreover, none of the arguments of our examples can be compared by this quasi-ordering.

9.4 Critical assessment of our novel specificity orderings

It has become clear in several discussions that the main obstacle for an acceptance of one of our relations $\lesssim_{\mathrm{CP1}}$ or $\lesssim_{\mathrm{CP2}}$ as a replacement for $\lesssim_{\mathrm{P3}}$ is the change this brings to Example 3 of Section 3: Some scientists working in the field have become used to the preference given by $\lesssim_{\mathrm{P3}}$ in this most popular example — so much that they now consider that preference a must. Note that the situation in Example 3 is actually most unstable under the two following aspects:

[47] Let us compare our specificity relations P3, CP1, CP2 with the "more conservative"-quasi-ordering by looking at our Corollaries 3, 5, and 8 in the context of Corollary 4. So let us assume $\mathscr{A}_1 \subseteq \mathscr{A}_2$. For the trivial case of $L_1 = L_2$, the argument $(\mathscr{A}_1, L_1)$ is quasi-smaller than the argument $(\mathscr{A}_2, L_2)$ for all of P3, CP1, CP2, and "more conservative". In case of $L_2 \in \mathfrak{T}_{\hat{\Pi}} \Rightarrow L_1 \in \mathfrak{T}_{\hat{\Pi}}$ and $\{L_1\} \cup \hat{\Pi} \vdash \{L_2\}$, again the argument $(\mathscr{A}_1, L_1)$ is quasi-smaller than the argument $(\mathscr{A}_2, L_2)$ for all of P3, CP1, CP2, but for "more conservative" it is the other way round, provided that we adopt the straightforward assumption that derivability is derivability w.r.t. the basic theory of $\hat{\Pi}$. Thus, P3, CP1, CP2 would all prefer $(\mathscr{A}, \mathsf{Mother}(\text{Claire de Long, Pierre Fermat}))$ to $(\mathscr{A}, \exists x.\mathsf{Mother}(x, \text{Pierre Fermat}))$, provided that we could express existential quantification.

1. The preference chosen by $\lesssim_{\mathrm{P3}}$ in Example 3 has justifications that are intuitive and valid, but are in general uncorrelated to specificity, such as the preference of conservativeness or the non-model-theoretic preference of defeasible derivations of shorter length. In particular in this example, such intuitive justifications easily contaminate the readers' intuition w.r.t. specificity. Moreover, as the arguments in Example 3 are not incomparable, but just equivalent according to $\lesssim_{\mathrm{CP1}}$, we can easily combine $\lesssim_{\mathrm{CP1}}$ lexicographically with another ordering, say "minimum in the ordering of the natural numbers, for all and-trees, of the maximal length of defeasible paths", and so recover the traditional preference of Example 3.
2. The situation of the example is chaotic in the sense that different preferences result from minor changes that may escape the readers' disambiguation.

   For instance, if we add the general rule of the example that precedes Example 3 (i.e. of Example 2), then the preference chosen by $\lesssim_{\mathrm{P3}}$ is chosen by $\lesssim_{\mathrm{CP1}}$ and $\lesssim_{\mathrm{CP2}}$ as well.

   Moreover, if we alternatively add bird(edna) as a fact, then we can embed the example injectively into Example 21 of Section 8.3.1, and then the preference chosen by $\lesssim_{\mathrm{P3}}$ is again chosen by $\lesssim_{\mathrm{CP1}}$ (whereas the arguments become incomparable w.r.t. $\lesssim_{\mathrm{CP2}}$).

Already the examples in Section 7 show, however, that $\lesssim_{\mathrm{P3}}$ almost always fails to prefer any argument in slightly bigger examples, not to speak of big ones. Indeed, $\lesssim_{\mathrm{P3}}$ can be considered a reasonable choice only if we restrict our considerations to tiny examples. Moreover, we presented good intuitive reasons for the failure of the preference of Example 3 in Example 9 of Section 6.6 (see also the pointers to further reasons in Note 28).

It is just too early for a further assessment, and the further implications of the contributions of this paper and the technical details of the operationalization of our correction of Poole's specificity will have to be discussed in future work.

Acknowledgments This research has been supported by the DFG (German Research Assoc.) grant Sto 421/5-1.

We would like to thank Matthias Thimm for his helpful discussions of Example 3. Moreover, we are deeply obliged to several anonymous reviewers of previous versions of this paper for their most profound critiques and suggestions for improvement, which have been a great help to improve the quality of presentation considerably.

To honor David Poole, let us end this paper with the last sentence of [22]:
This research was sponsored by no defence department.

References

1. Baral, C., De Giacomo, G., Eiter, T. (eds.): Proceedings of the 14th KR 2014 — International Conference on Principles of Knowledge Representation and Reasoning, Jul 20–24, as part of the Vienna Summer of Logic, Vienna, July 9–24, 2014, AAAI Press (2014). http://www.kr.org/index.php?page=proclist
2. Benferhat, S., Garcia, L.: A coherence-based approach to default reasoning. In: Gabbay, D., Kruse, R., Nonnengart, A., Ohlbach, H.J. (eds.) Proceedings of the 1st International Joint Conference on Qualitative and Quantitative Practical Reasoning, 1997, June 9–12, Bad Honnef (Germany), Springer, no. 1244 in Lecture Notes in Computer Science, pp. 43–57 (1997). doi:10.1007/BFb0035611
3. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artificial Intelligence **128**, 203–235 (2001). doi:10.1016/S0004-3702(01)00071-6. received Dec. 8, 2000
4. Besnard, P., Grégoire, É., Raddaoui, B.: A conditional logic-based argumentation framework. In: [20, pp. 44–56] (2013). doi:10.1007/978-3-642-40381-1_4
5. Chesñevar, C.I., Dix, J., Stolzenburg, F., Simari, G.R.: Relating defeasible and normal logic programming through transformation properties. Theor. Comput. Sci. **290**, 499–529 (2003). doi:10.1016/S0304-3975(02)00033-6. received Jan. 8, 2001; rev. Nov. 9, 2001
6. Clocksin, W.F., Mellish, C.S.: Programming in PROLOG. Springer (2003). 5th edn (1st edn.1981)

7. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**, 321–358 (1995). doi:10.1016/0004-3702(94)00041-X
8. Dung, P.M., Son, T.C.: An argumentation-theoretic approach to reasoning with specificity. In: Aiello, L.C., Doyle, J., Shapiro, S.C. (eds.) Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning, 1996, Nov. 5–8, Cambridge (MA), Morgan Kaufmann (Elsevier), Los Altos (CA), pp. 506–517 (1996)
9. Furbach, U., Glöckner, I., Pelzer, B.: An application of automated reasoning in natural-language question answering. AI Comm. **23**, 241–265 (2010)
10. Furbach, U., Schon, C., Stolzenburg, F., Weis, K.H., Wirth, C.P.: The RatioLog Project — Rational Extensions of Logical Reasoning. KI – Künstliche Intelligenz (German J of Artificial Intelligence), Springer **29**, 1–7 (2015). doi:10.1007/s13218-015-0377-9. published online June 05, 2015. Also in arXiv:1503.06087
11. Gabbay, D. (ed.): Handbook of Philosophical Logic. Kluwer (Springer Science+Business Media), 2nd edn (2002)
12. Gabbay, D., Woods, J.: Handbook of the History of Logic. Elsevier, North-Holland (2004)
13. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming, vol. 4, pp. 95–138. Cambridge University Press, Cambridge (2004)
14. Gelfond, M., Przymusinska, H.: Formalization of inheritance reasoning in autoepistemic logic. Fundamenta Informaticae **XIII**, 403–443 (1990)
15. Gillman, L.: Writing Mathematics Well. The Mathematical Association of America (1987)
16. Herbrand, J.: Recherches sur la théorie de la démonstration. PhD thesis, Université de Paris, no. d'ordre 2121, Série A, No. de Série 1252 — Imprimerie J. Dziewulski, Varsovie — Univ. de Paris. Also in Prace Towarzystwa Naukowego Warszawskiego, Wydział III Nauk Matematyczno-Fizychnych, Nr. 33, Warszawa (1930)
17. Kern-Isberner, G., Thimm, M.: A ranking semantics for first-order conditionals. [24, pp. 456–461] (2012). doi:10.3233/978-1-61499-098-7-456
18. Kowalski, R.A.: Predicate logic as a programming language. In: [25, pp. 569–574] (1974)
19. Lambert, J.H.: Neues Organon oder Gedanken über die Erforschung und Bezeichnung des Wahren und dessen Unterscheidung von Irrthum und Schein. Johann Wendler, Leipzig, Vol. I (Dianoiologie oder die Lehre von den Gesetzen des Denkens, Alethiologie oder Lehre von der Wahrheit) (http://books.google.de/books/about/Neues_Organon_oder_Gedanken_Uber_die_Erf.html?id=ViS3XCuJEw8C) & Vol. II (Semiotik oder Lehre von der Bezeichnung der Gedanken und Dinge, Phänomenologie oder Lehre von dem Schein)(http://books.google.de/books/about/Neues_Organon_oder_Gedanken_er_die_Er.htmlid=X8UAAAAAcAAj). Facsimile reprint by Georg Olms Verlag, Hildesheim, 1965, with a German introduction by Hans Werner Arndt (1764)
20. Liu, W., Subrahmanian, V.S., Wijsen, J. (eds.): Proceedings of the 7th International Conference on Scalable Uncertainty Management (SUM 2013), Washington (DC), Sept. 16–18, 2013, no. 8078 in Lecture Notes in Computer Science, Springer (2013)
21. Modgil, S., Prakken, H.: The ASPIC^{+} framework for structured argumentation: a tutorial. Argument & Computation **5**, 31–62 (2014). doi:10.1080/19462166.2013.869766
22. Poole, D.L.: On the comparison of theories: Preferring the most specific explanation. In: Joshi, A. (ed.) Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI), 1985, Aug. 18–25, Los Angeles (CA), Morgan Kaufmann (Elsevier), Los Altos (CA), pp. 144–147 (1985). http://ijcai.org/Past%20Proceedings
23. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In: [11, pp. 218–319] (2002)
24. Raedt, L.D., Bessière, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F., Lucas, P.J.F. (eds.): Proceedings of the 20th European Conference on Artificial Intelligence (ECAI), Aug. 27–31, 2012, Montepellier, France, no. 242 in Frontiers in Artificial Intelligence and Applications, IOS Press (2012). http://ebooks.iospress.nl/volume/ecai-2012
25. Rosenfeld, J.L. (ed.): Proceedings of the Congress of the International Federation for Information Processing (IFIP), Stockholm (Sweden), Aug. 5–10, 1974, North-Holland (Elsevier) (1974)
26. Simari, G.R., Loui, R.P.: A mathematical treatment of defeasible reasoning and its implementation. Artif. Intell. **53**, 125–157 (1992). received Feb. 1990, rev. April 1991
27. Stolzenburg, F., García, A.J., Chesñevar, C.I., Simari, G.R.: Computing generalized specificity. J. Applied Non-Classical Logics **13**, 87–113 (2003). doi:10.3166/jancl.13.87-113
28. Wirth, C.P.: Positive/Negative-Conditional Equations: A Constructor-Based Framework for Specification and Inductive Theorem Proving, Schriftenreihe Forschungsergebnisse zur Informatik, vol 31. Verlag Dr. Kovač, Hamburg, PhD thesis, Univ. Kaiserslautern, ISBN 386064551X. http://wirth.bplaced.net/p/diss (1997)

29. Wirth, C.P.: Shallow confluence of conditional term rewriting systems. J. Symb. Comput. **44**, 69–98 (2009). doi:10.1016/j.jsc.2008.05.005
30. Wirth, C.P.: Herbrand?s Fundamental Theorem in the eyes of Jean van Heijenoort. Logica Universalis **6**, 485–520 (2012). doi:10.1007/s11787-012-0056-7. received Jan. 12, 2012. Published online June 22, 2012
31. Wirth, C.P.: Herbrand's Fundamental Theorem: The Historical Facts and their Streamlining. SEKI-Report SR–2014–01 (ISSN 1437–4447), SEKI Publications, ii+47 pp., arXiv:1405.6317 (2014)
32. Wirth, C.P.: Herbrand's Fundamental Theorem — an encyclopedia article. SEKI-Report SR–2015–01 (ISSN 1437–4447), SEKI Publications, ii+16 pp., arXiv:1503.01412 (2015)
33. Wirth, C.P., Gramlich, B.: A constructor-based approach to positive/negative-conditional equational specifications. J. Symb. Comput. **17**, 51–90 (1994). doi:10.1006/jsco.1994.1004. http://wirth.bplaced.net/p/jsc94
34. Wirth, C.P., Stolzenburg, F.: David Poole's Specificity Revised. SEKI-Report SR–2013–01 (ISSN 1437–4447), SEKI Publications, ii+34 pp., arXiv:1308.4943 (2013)
35. Wirth, C.P., Stolzenburg, F.: David Poole's specificity revised. In: [1, pp. 168–177] Short version of [34] (2014)
36. Wirth, C.P., Siekmann, J., Benzmüller, Ch., Autexier, S.: Jacques Herbrand: Life, logic, and automated deduction. In: [12, Vol. 5: Logic from Russell to Church, pp. 195–254] (2009)
37. Wirth, C.P., Siekmann, J., Benzmüller, Ch., Autexier, S.: Lectures on Jacques Herbrand as a Logician. SEKI-Report SR–2009–01 (ISSN 1437–4447), SEKI Publications, Rev. edn. May 2014, ii+82 pp., arXiv:0902.4682 (2014)

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES:

Q1. Claus-Peter Wirth has been set as the corresponding author. Please check and advise if correct.
Q2. Please check affiliation if captured and presented correctly.
Q3. Journal requires Mathematics Subject Classfication (MSC) codes. Please provide.
Q4. Please check the suggested running page title if appropriate. Otherwise, please provide short running title.
Q5. Colored text in the body were changed to black. Kindly check and confirm if action taken was appropriate.
Q6. Footnotes 16 and 17 found in the image of Figure 1 were transpose instead in the figure caption as per Springers standard instruction. Kindly check and confirm if action taken was appropriate.
Q7. Please check Enunciation Example if captured and presented correctly.