

Against Nonmonotonic Logic (DRAFT)

Michael J. O'Donnell*

8 April, 1993

Abstract

The *monotonicity* property of inference in conventional formal systems of logic ($\mathbf{S} \vdash A$ and $\mathbf{S} \subseteq \mathbf{T}$ implies $\mathbf{T} \vdash A$) appears to prohibit certain crucial steps in practical intuitive reasoning. For example, reasoning by default (using assertions that are assumed to hold until explicitly contradicted) appears to be impossible in monotonic systems. Many crucial steps in practical reasoning certainly require nonmonotonic behaviors in inference procedures. But, I argue that it is highly misleading to describe these nonmonotonic behaviors in terms of nonmonotonic inference relations \vdash . Rather, inference procedures should use metatheoretic operators, with conventionally monotonic inference relations, to support nonmonotonic behavior.

1 Defeasible Reasoning Requires Nonmonotonicity

Most formal systems for logical reasoning are essentially definitions of a relation \vdash , called *logical inference*. When \mathbf{T} is a set of propositions expressed in the language used by such a formal system, and A is another such proposition, $\mathbf{T} \vdash A$ means that the formal system allows A to be inferred from hypotheses in \mathbf{T} . In the vast majority of formal systems studied by logicians, the relation \vdash is required to be monotonic in its left argument.

*Supported in part by NSF grant no. CCR-9016905

Definition 1 A logical inference relation \vdash is monotonic if and only if, for all sets of propositions \mathbf{S} and \mathbf{T} , and for all propositions A , if $\mathbf{S} \vdash A$ and $\mathbf{S} \subseteq \mathbf{T}$, then $\mathbf{T} \vdash A$.

Certain practical considerations appear to argue in favor of formal systems with *nonmonotonic* logical inference relations. I will sketch these practical considerations, but argue nonetheless that they are best accommodated by logical inference relations that are conventionally monotonic.

Practical reasoning often involves *defeasible* [?] (retractable) steps. In particular, there are often *default* assumptions [?] about typical cases, that are accepted until specific reasons to reject them are found. The example of concluding that a given bird flies until learning that the bird is a penguin has achieved chestnut status.

Example 1 (Penguins [?]) Consider a formal language with formulae expressing at least the following six propositions

$$\{ \text{Typical birds fly, Penguins do not fly,} \\ \text{George is a bird, George is a penguin,} \\ \text{George flies, George does not fly} \}$$

Given knowledge of the set of three propositions

$$\mathbf{T}_1 = \{ \text{Typical birds fly, Penguins do not fly, George is a bird} \}$$

it seems quite rational to suppose that George is in fact a typical bird, and conclude that George flies. That relation between a set of propositions and another proposition is often expressed as

$$\mathbf{T}_1 \vdash \text{George flies}$$

But, given the additional knowledge that George is a penguin, it seems irrational to conclude that George flies. Letting

$$\mathbf{T}_2 = \mathbf{T}_1 \cup \{ \text{George is a penguin} \}$$

this relation is expressed as

$$\mathbf{T}_2 \not\vdash \text{George flies}$$

But

$$\mathbf{T}_1 \subseteq \mathbf{T}_2$$

so the appropriate formal system of logic in which to perform practical reasoning about birds, penguins, and ability to fly seems to be nonmonotonic.

The desire to have a reasoning procedure that assumes typical conditions until finding evidence of a special case seems to prohibit monotonicity in the penguin example above. But, a closer analysis reveals more than one formal site for the nonmonotonicity of the procedure.

2 Two Formal Sites for Nonmonotonic Behavior

I believe that the penguin example, and many similar examples, provide a strong case for nonmonotonic behavior in a reasoning procedure. That is, there are certainly cases in practice where the discovery of new knowledge requires a retraction of old assertions. But, the explication of this nonmonotonic behavior can be attached to at least two different sites in a formal system of reasoning: the inference relation \vdash , and the formulae themselves. I will argue for the latter choice, but first I will reject some particular forms for doing so, in spite of their initial appeal.

In principle, nonmonotonic behavior of procedures using a monotonic inference relation may be explained by weakening the conclusions of inference. For example, instead of $\mathbf{T}_1 \vdash \textit{George flies}$, we might have

$$\mathbf{T}_1 \vdash \textit{George is a typical case} \Rightarrow \textit{George flies}$$

or

$$\mathbf{T}_1 \vdash \textit{Special case} \vee \textit{George flies}$$

Nonmonotonic behavior is produced by introducing a large or even infinite set of propositions, such as *George is a typical case*, as *default* assumptions, then removing them as soon as counterevidence appears. The extra clauses in conclusions serve as switches to enable or disable the real conclusions, such as *George flies*, by respectively including or excluding the default assumptions.

The basic idea of weakening conclusions to maintain monotonicity may be refined [?], but I believe that most reasoners rightly reject this complication

of the form of final conclusions. Since the normal mode of assertion of a proposition A in everyday life implicitly means something like, “*As well as can be determined, A holds,*” it seems inappropriate to introduce explicit formal disclaimers into every proposition. The best known formal systems for mathematical reasoning (the *Classical First-Order Predicate Calculus* heads the list, but the same holds for higher-order classical logics, set theories, and formalizations of intuitionistic or constructive reasoning) all treat the formal assertion of a proposition A as meaning something like, “ *A is absolutely true, with no possibility of error.*” But, nowhere in popular usage, not even in mathematical journals, is that formal notion actually practiced. If it were, Russell’s antinomy would have been a catastrophe rather than a stimulus to further research. So, I reject the weakening of conclusions by explicit formal disclaimers, since it forces a detailed accounting for assumptions that are made uniformly and universally in the practice of language, and thereby obscures the natural structure of practical assertions of propositions.

Rather than explaining nonmonotonic behavior by weakening the statements of the conclusions of inference, we need to strengthen the hypotheses. An obvious possibility is to add the same sorts of propositions that are used to weaken the conclusions, yielding

$$\mathbf{T}_1 \cup \{George\ is\ a\ typical\ case\} \vdash George\ flies$$

or the more transparent

$$\mathbf{T}_1 \cup \{George\ is\ not\ a\ flightless\ bird\} \vdash George\ flies$$

The trouble is that assertions of typicality, such as *George is a typical case*, do not represent direct intuitive judgements, but rather are themselves the results of sophisticated epistemological reasoning. Instead of begging the question of how one decides to assume *George is a typical case*, we should reduce that judgement to concepts that can be referred more directly to observations and intuition. The use of *George is not a flightless bird* avoids the explicit reference to typicality, but even more obviously begs the question of how we establish the hypothesis that *George is not a flightless bird*, which is essentially as hard as the final conclusion *George flies*.

Another way of augmenting the hypotheses is to use a formal operator *Search*, where for propositions A and sets \mathbf{T} of propositions, $Search(A, \mathbf{T})$

means that *a reasonably diligent search for known propositions relevant to A yields only T*. Then, the inference that *George flies* is represented by

$$\textit{Search}(\textit{George flies}, \mathbf{T}_1) \vdash \textit{George flies}$$

while

$$\mathbf{T}_1 \not\vdash \textit{George flies}$$

This approach captures the idea that the information in \mathbf{T}_1 alone does *not* allow us to conclude rationally that *George flies*; rather, the fact that a diligent search for knowledge bearing on *George flies* came up with nothing more than \mathbf{T}_1 is required to support the conclusion. *Search* is unusual in that it asserts something about the very inference procedure that is considering propositions built up from *Search*, so the operator is *metatheoretic* and *reflective* [?], in some ways analogous to the *negation as failure* used in certain implementations of Prolog [?].

Search can trigger nonmonotonic behavior, since the addition of new information to the knowledge base, or even the discovery of more information that was already in it, can change $\textit{Search}(A, \mathbf{T})$ from true to false. In the penguin example, once we discover the proposition *George is a penguin* among our knowledge, the metatheoretic proposition $\textit{Search}(\textit{George flies}, \mathbf{T}_1)$ becomes false, and the conclusion *George flies* can no longer be drawn, because the hypothesis required to infer it does not hold. $\textit{Search}(\textit{George flies}, \mathbf{T}_2)$ becomes true at this point, but it does *not* support the inference that *George flies*.

Further refinement of these ideas may show how to avoid the annoying repetition of *George flies* on both sides of \vdash , or may reveal that search keys have a different type than propositions, with some useful relations determining which search keys are relevant to inferring which propositions. One natural operator to consider is the *consistency* operator *Consistent* [?]. When \mathbf{T} is a set of propositions, $\textit{Consistent}(\mathbf{T})$ asserts that \mathbf{T} is internally consistent (noncontradictory). Using *Consistent*, we might approach the penguin problem by allowing

$$\mathbf{T}_1 \cup \{\textit{Consistent}(\mathbf{T}_1 \cup \{\textit{George flies}\})\} \vdash \textit{George flies}$$

In a reasonable system,

$$\textit{Consistent}(\mathbf{T}_1 \cup \{\textit{George flies}\})$$

holds, but

$$\textit{Consistent}(\mathbf{T}_2 \cup \{ \textit{George flies} \})$$

may not hold. In spite of the attractive conceptual simplicity of reasoning using *Consistent*, the huge cost of testing consistency, and the lack of apparent flexibility for dealing with conflicting defaults, suggest that we need an operator that refers somehow to the results of the limited searches for knowledge carried out by practical inference procedures, rather than the more abstract notion of consistency in a logical system. On the other hand, some sort of *resource-limited* test of consistency might be the right basis for introducing nonmonotonicity. Much further thought is required to determine whether expressions describing resource limitations need to be included as arguments to the operator, or whether they may be fixed or quantified out in a standard way.

In the next two sections, I argue that the use of a nonmonotonic logical inference relation \vdash is highly misleading, and *not* the appropriate way to support nonmonotonic behavior in a procedure for reasoning. Rather, I argue that nonmonotonic behavior should be supported by including, in the hypotheses of a logical inference, an explicit formal representation of the metatheoretic fact that certain propositions contain all of the knowledge relevant to a given problem discovered by a reasonably diligent search. I use the form *Search*(A, \mathbf{T}), where A is a proposition and \mathbf{T} is a finite set of propositions, to represent such metatheoretic assertions. I do *not* believe that this form is exactly the right one in practice, but it has enough of the essential elements to serve for the current comparative argument.

3 Two Interpretations of the Logical Inference Relation

The crux of my argument depends on choosing between two interpretations of the logical inference relation denoted by \vdash , one of which allows the relation to be nonmonotonic, and the other of which demands monotonicity. Roughly, $\mathbf{T} \vdash A$ means that, given knowledge of the propositions in \mathbf{T} , it is rational to infer A as well. Two different ways of making this rough idea more precise yield radically different results for practical reasoning.

Definition 2 *Let A be a proposition, and \mathbf{T} be a set of propositions.*

$$\mathbf{T} \vdash_L A$$

means that, whenever an agent knows each of the propositions in \mathbf{T} , the agent may rationally conclude that A holds as well (the subscript L stands mnemonically for “local”).

$$\mathbf{T} \vdash_G A$$

means that, whenever an agent knows each of the propositions in \mathbf{T} , and the agent knows that \mathbf{T} includes at least all the results of some reasonably diligent search for all available information relevant to judging the correctness of A , then the agent may rationally conclude that A holds as well (the subscript G stands for “global”).

The only distinction between \vdash_L and \vdash_G relevant to my discussion is in the different modes of asserting \mathbf{T} —as merely some set of known propositions in one case, or a set containing all known relevant propositions in the other case. The use of propositions rather than formulae, and sets \mathbf{T} rather than some other structures of propositions or formulae, makes no difference to the present discussion. By “knowledge” I mean reasonably reliable information that is relatively easily accessible to an agent. In particular, I do not assume that all knowledge is true. Some readers may wish to substitute “rational and informed belief” for “knowledge.”

Since knowing each of the formulae in a set \mathbf{T} implies knowing each of the formulae in every subset of \mathbf{T} , \vdash_L must be monotonic. \vdash_G is typically nonmonotonic, since as we make the set \mathbf{T} *larger*, the information that \mathbf{T} contains all available relevant knowledge becomes *weaker*, and therefore supports *fewer* inferences. In principle, logicians may choose to study any interesting relation between propositions and sets of propositions that they please. But, by focusing attention on an infelicitously chosen relation, and referring to it as “logical inference,” we may mislead those who are trying to apply formalisms to practical problems. An intuitively well grounded focus of attention may clarify thinking. In the next section, I analyze the concrete operations performed by a reasoning agent, and argue that \vdash_L is a useful relation for studying such agents, while \vdash_G is not.

4 The Case for Monotonic Logical Inference, with Nonmonotonic Procedures

The choice of an inference relation such as \vdash_L or \vdash_G as the basis for studying practical reasoning should depend on the following properties of a typical useful reasoning agent.

1. The agent has access to a huge set of propositions representing knowledge about the world, through some combination of queries to an internal database, observations of the world, and receipt of communications from other agents.
2. Access to available knowledge takes time, and consumes valuable resources, in a manner that is generally monotonic in some reasonable measure of the size of the set of propositions accessed.
3. The agent must do essentially all of its reasoning from small subsets of propositions selected from the available knowledge.
4. The agent must often interleave inference and access to knowledge—that is, an inference from knowledge accessed so far will often be required in order to determine how to search for further useful knowledge.

From these properties of a reasoning agent, it follows that the individual steps by which inferred propositions are added to current knowledge are of the form

- from some subset \mathbf{T} of current knowledge, find a proposition A such that $\mathbf{T} \vdash_L A$, and add A to current knowledge.

Property 4 requires \vdash_L here, instead of \vdash_G , since some inferences must be made before a reasonably diligent search is completed. \vdash_G essentially forces a reasoner into a two-phase procedure of search followed by inference.

In principle, propositions might be stratified hierarchically, so that whenever a judgement about the truth of A is required for a reasonably diligent search for all information relevant to B , then no judgement about B is required for a reasonably diligent search for all information relevant to A . Then \vdash_G might conceivably work, if we require \mathbf{T} to include all the propositions found by the latest search. That is, the two-phase procedure using \vdash_G may

be repeated to simulate interleaving of search and inference. But, experience doesn't suggest such a natural stratification, and it seems pointlessly expensive to require completion of a reasonably diligent search in the large number of cases where the inference does *not* require it semantically.

Certainly, \vdash_G is an interesting relation, and it is normally a stronger inference tool (*not* a logically stronger relation), in the sense that there are normally \mathbf{T} and A such that $\mathbf{T} \vdash_G A$, but $\mathbf{T} \not\vdash_L A$, while the reverse will not happen with conventional modes of assertion. So, in those cases where an appropriate search has been completed, and all of its results included in \mathbf{T} , an agent should be allowed to add A such that $\mathbf{T} \vdash_G A$. But, the agent must *notice* the fact that a diligent search is complete (this is necessary even if *all* inference steps use \vdash_G , since the agent must decide when to stop gathering members of \mathbf{T} , and start applying \vdash_G), so it seems natural to encode the act of noticing completion of the search as a metatheoretical proposition added to current knowledge. This leads to the encoding of

$$\mathbf{T} \vdash_G A$$

in a form such as

$$Search(A, \mathbf{T}) \vdash_L A$$

Certainly, the syntactic form for representing the proposition $Search(A, \mathbf{T})$ can be refined to be compatible with some useful data structure for noticing search completion, so this encoding need not reduce the efficiency of a reasoning procedure originally based directly on \vdash_G .

So, it appears that the information carried by \vdash_L is essential in the design of practical reasoning procedures, and that the information in \vdash_G may be encoded into \vdash_L by adding appropriate metatheoretic operators, which construct assertions that certain sets of propositions contain all of the results of certain searches, to our formal language. On the other hand, I doubt that \vdash_L can be defined efficiently and naturally from \vdash_G . An obvious candidate definition is

- $\mathbf{T} \vdash_L A$ if and only if $\mathbf{T}' \vdash_G A$ for all $\mathbf{T}' \supseteq \mathbf{T}$.

A naive application of this definition leads to an outrageously expensive implementation of \vdash_L . It seems unlikely that there is an efficient general implementation. Furthermore, it is not even clear that the definition is extensionally correct in practically interesting formal systems of logic. The

only if part probably holds in most reasonable systems. Monotonicity of \vdash_L requires that $\mathbf{T}' \vdash_L A$. When $\mathbf{T}' \vdash_L A$ we normally expect that $\mathbf{T}' \vdash_G A$ as well, since in the second case the hypothesis carries the information given directly by propositions in \mathbf{T}' *as well as* the information that \mathbf{T}' contains all results of some appropriate search. But, the *if* part seems to depend in a delicate way on the range of quantification of extended sets \mathbf{T}' of propositions, which varies depending on the generality of the formal language in use.

I do *not* claim that the particular syntactic form of $\text{Search}(A, \mathbf{T})$ is the right one for practical reasoning. But, this form illustrates some of the crucial properties of the right solution.

1. Inferences using default reasoning must have *metatheoretic* hypotheses, since the basis for using a default assumption is the inability to discover a counterargument, rather than some direct observation about the world outside of a reasoning agent.
2. Some sort of search parameter must be involved in such metatheoretic hypotheses, since all practical searches are guided by some description of a goal, and the results of search depend critically on this description. The A in $\text{Search}(A, \mathbf{T})$ is a very naive presentation of this search parameter.
3. Some sort of description of the results of search must be involved in such metatheoretic hypotheses, since it is the absence of a counterindication in those results that enables the default assumption. The \mathbf{T} is a very naive presentation of the results of the search.
4. Although the extensional contents of the knowledge base available to a reasoning agent determines the truth or falsehood of such metatheoretic hypotheses, the contents of that knowledge base must *not* be mentioned explicitly in the formal presentation of the hypotheses, since it is too big. The whole point of a formal logical language is to represent the information actually manipulated by a reasoning agent, which should normally be much smaller than the world about which the agent is reasoning. For comparative discussion of the results of reasoning from different knowledge bases, some description of the knowledge base may be treated as an additional parameter to the logical inference relation \vdash , but it must be understood that the reasoning agent has no direct

access to it, only to the formal propositions that result from searching it. In particular, useful definitions of \vdash must not depend on qualities of the knowledge base other than those derived by the search procedures used by the reasoning agent. If several different knowledge bases are available simultaneously to a single agent, the *name* of a knowledge base to be searched may be modelled as part of the search parameter mentioned in item 2, but the *contents* of the knowledge base must not be an explicit parameter.

5 The Hard Problem Remains

Of course, arguments of the sort advanced above do not solve any hard technical problems. The hard problem to solve in automating practical reasoning is the problem of reacting to *updates* (changes in knowledge) [?]. That problem is essentially the same at a technical level, whether we use a monotonic inference relation \vdash_L or a nonmonotonic \vdash_G . In conventional monotonic systems, without metatheoretic operators, the addition of new knowledge leads only to the *addition* of new inferences. The hard part of the update problem is how to handle *retraction* of previously accepted knowledge. There is no clear consensus even on the proper form in which to express retractions.

Both nonmonotonic inference relations, such as \vdash_G , and metatheoretic operators, such as *Search*, seem to add a complication to the problem of changing knowledge, since they both lead to cases where *added* knowledge requires conclusions to be *retracted*. I conjecture, though, that once the basic problem of managing retraction of knowledge is solved, the extra interaction of additions and retractions introduced by nonmonotonicity will become much clearer. I propose that, by representing nonmonotonic reasoning behavior through metatheoretic operators, such as *Search*, with the monotonic inference relation \vdash_L , we can improve the flexibility and transparency by which future breakthroughs in the update problem are applied to a wide range of interesting domains of reasoning.

While the 4 properties of useful reasoning agents mentioned in Section 4 support the use of formal systems that are conventional to the extent of being monotonic, they by no means support the particular choices of classical or intuitionistic reasoning. In fact, an additional property of reasoning agents argues strongly against such choices:

5. Some of the knowledge provided to the agent is wrong, and possibly even contradictory.

Both classical and intuitionistic logic trivialize in the presence of contradictory information—*everything* follows logically from any contradiction, and the mere fact that classically or intuitionistically $\mathbf{T} \vdash A$ does *not* make it safe for a reasoning agent to conclude A given knowledge of each proposition in \mathbf{T} . Even in the absence of contradiction, classical and intuitionistic logic seem to be too sensitive to errors, due to their conceptual foundation on the assertion of absolute truth rather than of rational belief based on fallible information, although I can find no rigorous discussion of this sensitivity in the literature.

An obvious fix is to have the agent test the consistency of \mathbf{T} before applying logical inference. Unfortunately, consistency checking ranges from intractably expensive to fundamentally impossible, so the time and resource for consistency checking would dominate that for real reasoning steps. Rather, property 5 requires the use of *paraconsistent* formal systems [?], such as *relevance logics* [?] instead of classical or intuitionistic formal systems. Further research is required to illuminate the choice of a particular paraconsistent formal system for a particular reasoning agent. The right paraconsistent formal system can reduce the harm done by errors and contradictions in the hypotheses used by a reasoning agent, allowing useful reasoning to go on during the typically long time between the introduction of an error or contradiction and its detection. But, they still do not address the hard problem of retracting information when an error is finally detected.