Resource-Bounded Reasoning and Paraconsistency

M. Allen
Department of Philosophy
University of Pittsburgh
Pittsburgh, PA, U.S.A.

R. E. Jennings
Laboratory for Logic & Experimental Philosophy
Department of Philosophy
Simon Fraser University
Burnaby, BC, Canada

Abstract ***Paraconsistent logic formalizes reasoning from inconsistent information. The preservationist variety of this logic concerns itself with forms of reasoning that preserve properties of information other than those, like overall satisfy and consistency, preserved by classical forms. The study of computation and information processing in the presence of inconsistency has utilized paraconsistent logics, although not generally of the preservationist kind. We argue that situations involving bounded resources for computation can give rise to situations that naturally call for preservationist paraconsistent reasoning. The point is illustrated with reference to memory- and time-limited systems, yielding approximate answers to database queries.***

Keywords: inconsistency, paraconsistent logic, preservationist logic, resource-bounded reasoning, databases

1 Introduction

The use of logic to represent reasoning, or at least correct reasoning, can introduce a few potentially problematic idealizations. First, it is common, for reasons normative and practical, to represent reasoning as dealing always with consistent information, despite the rarity of such strict standards of rational hygiene among actual reasoners. Second, logic tends to bring with it notions of *closure*. Reasoners, for instance, are pictured as believing the deductive closure of their beliefs—everything that follows logically from those beliefs—despite the fact that this closure is generally an infinite collection, and reasoners are usually of finite capacity.

While some AI researchers, like Wooldridge [1], have accepted these idealizations, others, like Russell and Wefald [2], have argued that representations of reasoning, natural or mechanical, must take account of limited resources and bounds on computational power. Too, the recognition that knowledge and information are often inconsistent has led to interest in logics that are non-classical. In fact, there are important connections between these ideas. Reasoners with imperfect, bounded capacities may not be in any position to ensure that all information with which they deal is consistent; some inconsistencies will be obvious to them, while others may be practically inaccessible. In such cases, representations of reasoning need to respect the difference between what is possibly of interest to the reasoner, and what is beyond the bounds of their capacities.

2 Paraconsistent logic

In its most general form, paraconsistent logic is "the study of those logics which formalize nontrivial inconsistent theories" (Da Costa and Wolf [3]). Whereas inconsistency leads to inferential profligacy in classical logic, and anything follows from an inconsistency, paraconsistent logics restrict the consequences of at least some inconsistencies. Priest, Routley, and Norman

[4] provide a survey of a number of the approaches that fall under this general description. Among computer scientists, one of the most influential such proposals has been the 4-valued logic of Belnap [5], in which a computer assigns values to information in accord with whether it has been told that the information in question is true or false. Four possibilities arise since the computer may be told that certain pieces of information are *both* true and false, and may be told *neither* thing of other pieces. A scheme like this is employed in the paraconsistent logic programming of Blair and Subrahmanian [6], to provide semantics for inconsistent knowledge-bases. Multi-valued models mean that meaningful queries are possible, even where classical models of the knowledge-base are unavailable.

2.1 Preservationist approaches

Multi-valued paraconsistency is common, but is only one alternative. This approach basically extends the traditional idea that logical reasoning preserves the truth of sets of premises, by extending the semantics to satisfy inconsistencies—things that were once unsatisfiable. “Good” forms of reasoning are still those that preserve satisfiability, even if satisfaction is now something other than it was in the classical context. *Preservationist logic* takes a broader view of what may be preserved of sets of sentences. Preservationists may want to preserve what satisfiability there is, but, especially where there is no such thing on offer, will seek to preserve other things as well.

Preservationist inquiry proceeds in one of two directions: on the one hand, it examines various of the properties possessed by information, and looks for logical means of preserving them; on the other hand, it considers consequence- and implication-relations, and asks what properties they might preserve. An early example of the approach is found in Schotch and Jennings [7], who focus on the preservation of “level of coherence,” measured by the number of times a set need be partitioned to yield classically consistent subparts. Other work focusses on other properties. Thorn [8] deals with preserving “dilution” of inconsistency, marked by the size of the smallest inconsistent subset. Brown [9] and Allen [10] measure levels of ambiguity for sets of sentences, and examine logics that preserve minimal levels of same.

In each case, a *measure* is established. Some sets possess the measure property to a greater or lesser degree than others, and it is established which of these conditions is more desirable. The preservationist logician then works to establish means of preserving the desirable, and fending off the undesirable. While classical logic preserves the satisfiability of a set of sentences, and holds off inconsistency, so long as the set is already satisfiable and entirely consistent, a coherence-level-preserving logic preserves a certain *degree* of consistency, so long as a set already possesses that much. Recognizing the variousness of the properties of information that one might care to maintain, preservationism expands the logical horizon. Jennings, Chan, and Dowad [11] argue that this approach provides a broad understanding of “inference,” of interest to computer scientists so far as they are concerned with modes of reasoning in specialized domains, and so with a range of distinct properties of information.

3 Bounded reasoning and inconsistency

Logical models of reasoning often build in the presumption that reasoners are infinite in capacity; however, Real-world analogues of reasoning, of which computation is an example, will involve reasoners with finitely bounded capacities. Indeed, some contexts, particularly where machine reasoning is concerned, impose strict limits on capacity, limits well below even what might be desired. Systems operating with restricted memory or limited time, for example, may not have the resources to provide strictly optimal performance. Given such bounds, useful forms of reasoning may well differ from those associated with the limitless case. Research in *resource-bounded reasoning* studies such situations. Generally, a resource-bounded reasoner will need to perform, not perfectly, but as well as can be expected. Standards of success are set, and systems are designed so that performance increases along with availability of resources. As Zilberstein [12] has it, it still makes sense to speak of conformity to rational standards in the presence of limits—rather than identify rationality with optimality *tout court*, it is identified with optimality within bounds.

Work on knowledge-base reasoning has taken account of the problems posed by both inconsistent information and bounded performance. As noted, paraconsistent logic programming has provided multi-valued semantics for inconsistent knowledge-bases. Too, Vrbsky and Liu [13] have developed a system for handling approximate database queries that gives inexact answers, but tends toward correctness as time and other resources increase. It is interesting to combine these concerns, since real-world systems are likely both to be limited in capacity, and to be dealing with inconsistent information. While one approach provides a paraconsistent semantics for an entire knowledge-base where inconsistency arises, systems with limited resources may never deal with the knowledge-base as a whole. In such cases, only certain sorts of inconsistency are salient. Where a system has the resources to examine an entire database together, and to perform all logical operations on it, the presence of inconsistency may well be dealt with using one of the multi-valued paraconsistent alternatives. In such a system, we generally have the full range of usual rules for inference and for manipulating expressions of the language. A typical rule is *adjunction*: conjunctions of any two expressions implied by the knowledge-base are also implied by the knowledge-base. In a system that can only access a limited portion of the knowledge-base at any one time, however, such a rule may not be realistic. While an accessible portion of the information implies some expression A, and another accessible portion implies another expression B, there may be no such single portion that implies both A and B; in such a case, a general adjunction-introduction rule no longer reflects the sorts of outputs actually produced by the limited system. Similarly, not all inconsistencies are equal—some are more readily accessible than others. To use a toy example, consider the the simple sets $\{ p, p \to q, \neg q \}$ and $\{ p, p \to q, q \to r, \neg r \}$. While both are classically inconsistent, inconsistency is more easily accessible in the first case than the second, since the first only uses three sentences to generate the contradiction, whereas the second uses four. Similar cases arise on a larger scale for systems that access only a portion of a knowledge-base at a time; certain inconsistencies may be "visible" only at scales greater than those available to the system.

4 Level and dilution preservation

Recognizing the differences between varieties of inconsistency just mentioned, we are interested in preservationist forms of reasoning that take them into account. Two properties of interest are *level of coherence* and *dilution of level.*

4.1 Level of coherence

Level of coherence (or simply "level") for a set of sentences is the size of the least partition (if there is any such) of the set into classically-consistent parts. We define a partition as follows: for any set Σ, the set of sets $\pi = \{ a_1, a_2, \ldots, a_n \}$ is a *partition* of Σ, written $\mathrm{Part}(\pi, \Sigma)$, just in case each "cell" $a_i \in \pi$ is a subset of Σ, the cells do not intersect, and together they cover Σ. That is, $\mathrm{Part}(\pi, \Sigma)$ iff:

1. $\forall a_i \in \pi,\ a_i \subseteq \Sigma$.

2. $\forall a_i, a_j \in \pi,\ (a_i \neq a_j) \Rightarrow (a_i \cap a_j) = \emptyset$.

3. $\bigcup \{ a_i \mid a_i \in \pi \} = \Sigma$.

We can now define the collection of *n-partitions* of a set Σ, written $\Pi_n(\Sigma)$, as the set of partitions of Σ of size n: $\forall n\,(1 \leq n \leq \omega)$,

$$\Pi_n(\Sigma) = \{\, \pi \mid \mathrm{Part}(\pi, \Sigma) \;\&\; \|\pi\| = n \,\}$$

The level of coherence for set of sentences Σ is defined as the minimal value n such that there is some partition $\pi \in \Pi_n(\Sigma)$ whose cells are all classically consistent. We express consistency using the notion of implication and an arbitrary absurdity, $\bot$. For any sentence α, $\bot$ implies α, written $\bot \vdash \alpha$. Similarly, a set Σ is classically inconsistent if and only if it $\Sigma \vdash \bot$; consistency is expressed by denying the implication, $\Sigma \nvdash \bot$. We say that the level of coherence of Σ is n, written $l(\Sigma) = n$, if and only if n is the least value such that

$$\exists \pi \in \Pi_n(\Sigma) : \forall a \in \pi,\ a \nvdash \bot.$$

If $\bot \in \Sigma$, then there does not exist any such value of n, and we say Σ has no measurable level, assigning $l(\Sigma) = \infty$, an arbitrarily high value. If Σ is already classically consistent, then $l(\Sigma) = 1$. Finally, the level of Σ, if measurable, is always less than or equal to the size of Σ itself, since for any value n, $1 \leq n \leq \omega$, it takes at least n members to create a set that can be partitioned into n cells.

Examples: The set $\Sigma = \{\, p, \neg p, q \,\}$ has $l(\Sigma) = 2$. The set $\Delta = \{\, p \wedge q, \neg p \wedge q, p \wedge \neg q \,\}$ has $l(\Delta) = 3$.

4.2 Dilution of level

We now define a companion notion, the dilution of level for Σ. For set Σ, and measurable level n, dilution of level-n of Σ, written $d_n(\Sigma)$, is the size of the smallest subset of Σ having level n (if any such exists). That is, $d_n(\Sigma) = m$ if and only if m is the least value such that

$$\exists \Sigma' \subseteq \Sigma\!:\ l(\Sigma') = n \;\&\; \|\Sigma'\| = m.$$

Note that for any value of n, $1 \leq n \leq \omega$, and any set Σ, if $l(\Sigma) = n$, then for all n', $1 \leq n' \leq n$, there exists some m such that $d_{n'}(\Sigma) = m$. For values $n > l(\Sigma)$, $d_n(\Sigma)$ is undefined.

Examples: The set $\Sigma = \{\, p, p \rightarrow q, \neg q \,\}$ has $d_2(\Sigma) = 3$. The set $\Delta = \{\, p \wedge q, \neg p \wedge q, p \wedge \neg q \,\}$ has $d_3(\Delta) = 3$ and $d_2(\Delta) = 2$.

As the first of these examples shows, dilution of level-n can be greater than n itself. Indeed, n provides but a lower bound. For any measurable value, $d_n(\Sigma) \geq n$. Furthermore, all possible combinations of measurable level and dilution exist; for any measurable level n, and any $m \geq n$, some Σ exists such that $d_n(\Sigma) = m$. Thorn [8] provides a schema for generating sample instances. Measurable dilution values are also ordered in accordance with level. For any Σ, for measurable levels n and n', if $n' < n$ then $d_{n'}(\Sigma) < d_n(\Sigma)$.

4.3 Preserving level and dilution

There are any number of strategies preserving level of coherence, but perhaps the simplest also involves set-partitions. If $l(\Sigma) = n$, we can preserve that value by adding back to Σ those sentences that follow classically from every n-partition. That is, Σ *n-forces* α, written $\Sigma \,[\!\vdash_n \alpha$, if and only if

$$\forall \pi \in \Pi_n(\Sigma),\ \exists a \in \pi\!:\ a \vdash \alpha.$$

Clearly, n-forcing preserves level n. If $l(\Sigma) = n$, and Σ n-forces α, then $l(\Sigma \cup \{\alpha\}) = n$ as well. This follows since, if $l(\Sigma) = n$, then there exists some n-partition of Σ, call it $\pi^\star$, into n classically-consistent parts. If α is n-forced by Σ, then α follows from some cell of every n-partition of Σ, including $\pi^\star$. If cell $a^\star \in \pi^\star$ classically implies α, then we know that $(a^\star \cup \{\alpha\})$ must be classically consistent. So, there exists a partition of $(\Sigma \cup \{\alpha\})$ into n consistent cells, namely the n cells of the partition $\pi^\star$, with α added into $a^\star$. That is, $l(\Sigma \cup \{\alpha\}) = n$, and n-forcing preserves level n. Finally, note that if $l(\Sigma) = 1$, and the set is already classically consistent, n-forcing is equivalent to classical implication: $\Sigma \,[\!\vdash_1 \alpha \Leftrightarrow \Sigma \vdash \alpha$.

In the case of dilution of level, the point is to keep dilution from *decreasing*. A dilution-preserving strategy will ensure that anything added to Σ will do nothing to make the inconsistency of Σ more concentrated. In this connection, note that adding α to Σ decreases $d_n(\Sigma)$ only by *increasing level* for some proper subset. That is, if $d_n(\Sigma \cup \{\alpha\}) < d_n(\Sigma)$ then there is some m such that

$$d_n(\Sigma) = m \; \& \; \exists \Sigma' \subset \Sigma: \\ \|\Sigma'\| \leq (m-2) \; \& \; l(\Sigma') < l(\Sigma \cup \{\alpha\}).$$

To see this, note that $d_n(\Sigma \cup \{\alpha\}) < m$ only if some subset $\Sigma' \subset (\Sigma \cup \{\alpha\})$ has level n and size less than m; that is, $\|\Sigma'\| \leq (m-1)$. Furthermore, we know that $\alpha \in \Sigma'$, since otherwise the n-dilution of Σ would have been lower than m in the first place. So, we know that $\|(\Sigma' - \{\alpha\})\| \leq (m-2)$. As well, we know that $l(\Sigma' - \{\alpha\})$ must be less than n, again because we already have that $d_n(\Sigma) = m$, and the size of the subset is less than m.

Given these facts, it follows that for any set Σ such that $d_n(\Sigma) = m$, we can preserve the value for n-dilution by preserving the level of every subset with less than m members. In fact, since the the level of a set is always greater than or equal to the level of any of its subsets, it suffices simply to preserve the level of every $(m-1)$-membered subset; and, since we already have a strategy for preserving level of sets, we easily arrive at a way of preserving dilution of level. We know, then, that α does not decrease the dilution of level n for any Σ such that $d_n(\Sigma) = m$ if α is $(n-1)$-forced by every $(m-1)$-membered subset of Σ. That is, if

1. $d_n(\Sigma) = m$, and
2. $\forall \Sigma' \subset \Sigma$, $\|\Sigma'\| = (m-1) \Rightarrow \Sigma' [\!\vdash_{n-1} \alpha$,

then $d_n(\Sigma \cup \{\alpha\}) = m$, as well.

5 Applications

We can apply these results to problems faced by bounded reasoners in the context of inconsistent knowledge-bases. We have devised a resource-bounded reasoning procedure for use with a system with strictly limited, and fixed memory. The sort of system in question has been tested on randomly-generated knowledge-bases, using a simple database language, in which each expression is a disjunctive Horn-like sentence. Upon request, the system queries the database, but can only access sub-portions of the database at a time; this portion is determined in advance as a bound p on the number of sentences from the knowledge-base that can be processed together at any one time. In addition, the sentences which the system can process are strictly limited in length, again by fixed memory constraints. In this system, then, arbitrary adjunctions of sentences are not possible, since the system cannot process many of the expressions resulting from application of such a rule. Thus, the system is also strictly bound, again by value p, as to the salient concentration of inconsistency. So long as a particular level of coherence is concentrated enough to be evident in some collection of sentences of size p or less, the system can ensure that the given level does not become any more concentrated. In certain measurable respects, then, the system works to ensure that its knowledge-base does not become any more inconsistent than it already is: queries are answered in the affirmative only if the response does not decrease the dilution of a given level of coherence.

Because the system operates also under possible time-constraints, it performs its query-response function by way of *anytime algorithms.* These sorts of algorithms, as detailed in Zilberstein [14], can yield results at any time during operation. While results improve over time, approximate answers are available at any time after passage of a short initial interval. In our system, the computer first runs the query on the initial p sentences in its database, using resolution methods. For sake of illustration, suppose that we wish accessible segments of our database Δ to be entirely consistent. That is, we want that $d_2(\Delta) > p$: all inconsistent segments of the database are larger than p in size. Results of the prior sections indicate that this condition can be maintained if every segment of Δ that is of size p or smaller be kept at level one. Since our computer can access segments of size p, it suffices to 1-force from each of these segments, which is to say take what follows by classical methods from each and every one of them. However, we do not know in advance whether or not the dilution of level for our database actually is greater than p. This being so, we require that the computer try to answer queries in such a way that it *would* preserve the requisite dilution, so long as the right conditions already existed; at the same time, and so far as it can, it checks whether the inconsistency present already falls outside of acceptable bounds.

The system, as mentioned, can be interrupted at any point in its progress through the database Δ. In the meantime, it runs the query on each p-membered segment of Δ in turn, first determining whether the query-expression is implied by the segment, and then determining whether the negation of the expression is implied by the same segment. If the answer to the first question is yes, this answer is stored as the final result of the query, to be returned whenever the program terminates or is interrupted. As time goes on, and more and more of the segments of the database are queried, the answer tends ever more gradually toward correctness. The current degree of correctness can be measured straightforwardly in terms of the probability that the answer is correct, given how many of the p-membered segments of the database have been examined; as required, this value increases monotonically with time of processing. If at any point some segment does not imply the query-expression, then the program will cease operation, returning a negative response to the query, since we are only here interested in expressions implied by every one of the p-membered subsets, in order to retain a dilution equal to $p + 1$. (There is a less strict strategy that also results in the same preservation, that only requires one such subset to imply the query-expression, but we are not here directly concerned with that idea.) Finally, if the machine ever derives both the query-expression and its negation from a single p-membered subset of the database, then it also halts, returning the result that, in this case, the database already has a dilution of inconsistency *below* the value p; it is then up to the user to decide whether or not the condition is tolerable, and revise the database if necessary. Again, the performance of this detection function also improves monotonically over time; as the machine handles more and more of the requisite subsets of the database without finding an impermissible degree of inconsistency, it becomes more and more likely that the database as a whole actually has the demanded level and dilution.

We only mean to sketch the behaviour of the system here, but we note some important features. First, while the example just considered is one in which we insist upon consistency of all p-membered subsets of the database, other possibilities are open to us. For any level of coherence less than or equal to p, the size of our working memory, and for any possible dilution of that level, the system can work to preserve the requisite dilution of level, by employing the right sort of forcing operation over the appropriate range of subsets of each segment. In each case, if the level is set to be $n \leq p$, then we can also fix dilution of level at any value m, $n \leq m \leq p + 1$; by employing $(n-1)$-forcing on every $(m-1)$-membered subset of the database, we ensure that $d_n(\Delta) > m$. This opens the door for a wide range of possibilities when it comes to processing of inconsistent information. Standards for tolerable inconsistency can be set, and maintained, up to a degree strictly bounded by the size of working memory p, and these standards can be met up to a degree related to the amount of time available for processing. At the same time, we note that the amount of processing required can depend rather heavily on the differences between the size of the available memory p and the size of the database as a whole. If p is significantly less than the size of the whole base Δ, and the size of Δ is quite large, computation of an exact answer to the query and detection of impermissible inconsistency become intractable, since the machine must always process $\binom{q}{p}$ subsets, where q is the size of the database as a whole. Similarly, as the potential level of the p-membered subsets increases, the problem of combinatorial explosion again rears its head, as n-forcing also requires the computer to survey ever-increasing numbers of partitions. Our anytime system has the advantage of always being able to return at least approximate answers, but the horizon on which the correct answers are met tends to recede rapidly. The relation between these various values and measures, and the resources necessary for dealing with them, demands further study.

6 Conclusions

Research in logics that preserve various properties of sets of sentences has direct connection to research in machine reasoning. By studying ways in which we can measure of inconsistency in information, and preserve same, we find new strategies for mechanical reasoning in the presence of such inconsistency. So far as dilution of level corresponds to the degree to which inconsistency can become salient to a reasoner, research into the property and its preservation is of particular interest in connection with questions of bounded reasoning.

References

[1] Michael Wooldridge. *Reasoning About Rational Agents.* MIT Press, Cambridge, Massachusetts & London, England, 2000.

[2] Stuart Russell and Eric Wefald. *Do the Right Thing: Studies in Limited Rationality.* MIT Press, Cambridge, Massachusetts & London, England, 1991.

[3] Newton C. A. da Costa and R. G. Wolf. Studies in paraconsistent logic I. *Philosophia*, 9:189–217, 1980.

[4] Graham Priest, R. Routley, and J. Norman, editors. *Paraconsistent Logic: Essays on the Inconsistent.* Philosophia Verlag, Munich, 1989.

[5] Nuel D. Belnap, Jr. A useful four-valued logic: How a computer should think. In Alan Ross Anderson, Nuel D. Belnap, Jr., and J. Michael Dunn, editors, *Entailment: The Logic of Relevance and Necessity*, volume II, pages 506–541. Princeton University Press, Princeton and Oxford, 1992.

[6] Howard A. Blair and V. S. Subrahmanian. Paraconsistent logic programming. *Theoretical Computer Science*, 68:135–154, 1989.

[7] Peter K. Schotch and R. E. Jennings. On detonating. In Priest et al. [4], pages 306–327.

[8] Paul D. Thorn. The normative character of interpretation and mental explanation. Master's thesis, Simon Fraser University, Burnaby, British Columbia, Canada, 1998.

[9] Bryson Brown. Yes, Virginia, there really are paraconsistent logics. *Journal of Philosophical Logic*, 28:489–500, 1999.

[10] M. Allen. A paraconsistent-preservationist treatment of a common confusion concerning predicate extensions. In B. Brown and J. Woods, editors, *Logical Consequences (Proceedings of the SEP Annual Conference, 1999).* Hermes Science Publishing, forthcoming.

[11] R. E. Jennings, C. W. Chan, and M. J. Dowad. Generalised inference and inferential modelling. In *Proceedings of the 12th International Conference on Artificial Intelligence*, pages 1046–1051, 1991.

[12] Shlomo Zilberstein. Models of bounded rationality. Unpublished (available at *anytime.cs.umass.edu*). Presented at the AAAI Fall Symposium on Rational Agency, Cambridge, Massachusetts, 1995.

[13] Susan V. Vrbsky and Jane W. S. Liu. APPROXIMATE—A query processor that produces monotonically improving approximate answers. *IEEE Trans. on Knowledge and Data Engineering*, 5:1056–1068, 1993.

[14] Shlomo Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, pages 73–83, 1996.