Report Number: WUCS-89-41

1989-09-01

Ampliative Inference, Computation, and Dialectic

R. P. Loui

There are three theses here: • Non-computationally conceived inference merely expands notation. This includes induction as well as deduction, and thus both deserve the adjective non-ampliative. Deriving entailments merely expands shorthand. All of the familiar formalisms for reasoning do just this. • There now exist examples of formalism for reasoning that do something else. They are deliberative, and to say in what way they are deliberative requires reference to the process through which they compute their entailments. • The original ampliative/non-ampliative terminology best survives as referring to this new distinction. Viewed formally, all other attempted distinctions either presume deduction... **Read complete abstract on page 2.**

Ampliative Inference, Computation, and Dialectic

R. P. Loui

Complete Abstract:

There are three theses here: • Non-computationally conceived inference merely expands notation. This includes induction as well as deduction, and thus both deserve the adjective non-ampliative. Deriving entailments merely expands shorthand. All of the familiar formalisms for reasoning do just this. • There now exist examples of formalism for reasoning that do something else. They are deliberative, and to say in what way they are deliberative requires reference to the process through which they compute their entailments. • The original ampliative/non-ampliative terminology best survives as referring to this new distinction. Viewed formally, all other attempted distinctions either presume deduction to be privileged, or else fail to separate inference that actually tells us something new from inference that simple rehashes what has already been represented.

AMPLIATIVE INFERENCE, COMPUTATION, AND DIALECTIC

R. P. Loui

WUCS-89-41

September 1989

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

ABSTRACT

There are three theses here:

1. Non-computationally conceived inference merely expands notation. This includes induction as well as deduction, and thus both deserve the adjective non-ampliative. Deriving entailments merely expands shorthand. All of the familiar formalisms for reasoning do just this.

2. There now exist examples of formalisms for reasoning that do something else. They are deliberative, and to say in what way they are deliberative requires reference to the process through which they compute their entailments.

3. The original ampliative/non-ampliative terminology best survives as referring to this new distinction. Viewed formally, all other attempted distinctions either presume deduction to be privileged, or else fail to separate inference that actually tells us something new from inference that simply rehashes what has already been represented.

*This paper will be included in AI and Philosophy edited by R. Cummins and J. Pollock, MIT Press, 1990.

Ampliative Inference, Computation, and Dialectic*

R. P. Loui
Dept. of Computer Science and
Dept. of Philosophy
Washington University
St. Louis, MO 63130

September 23, 1989

1

What makes inference ampliative?[1] Traditionally, deductive inference has been considered non-ampliative, while induction and analogy have been ampliative. This distinction is threatened by AI's view of representation and reasoning.

Representing knowledge in a language is a peculiar responsibility. Inference rules applied to what is explicitly represented in a language reveal what is implicitly represented. This terminology (etched in AI minds by H. Levesque)[2] is no accident. In the AI view, the representer of knowledge is responsible for both the explicit and the implicit. That is, in representing a set of sentences explicitly in a language, it must be the intention to represent the implicit knowledge too. Inept use of language is discounted as a possibility. All language users are assumed facile.

Against this backdrop, two anima conspire against the ampliative–non-ampliative distinction. They are the advent of logics of defeasible reasoning (or non-monotonic reasoning), and the extreme formality with which AI requires that inference systems be viewed. The rise of defeasibility in the formalization of inference derides the distinction between ampliative and non-ampliative because defeasible reasoning combines inductive and deductive inference. If indeed non-monotonic reasoning is a bit like non-ampliative deduction, especially in form,

*for J. Pollock and R. Cummins, *AI and Philosophy*, MIT Press, 1990.

[1] I am indebted to D. Israel for bringing this question to my attention. Apparently, we should thank a W. Johnson for raising it in the first place.

[2] Levesque used the explicit/implicit distinction for different purposes. Doyle's use of the distinction between constructive and manifest belief is closer to what I have in mind, but I prefer the phrase "implicit commitment" to the phrase "constructible commitment." Could history be altered, "manifest" would be a better term for Levesque, and "explicit" would be the better term for Doyle.

but also a lot like ampliative induction, then which is it: ampliative or not? Like induction, it makes guesses that allegedly go beyond the evidence. Like deduction, the license to produce its conclusions is written plainly for all to see.

AI views inference with extreme formality. A system of inference is a programming language. One language is as good as another, though some are more expressive, and there are unmistakable differences in convenience. Change the logical language and one simply changes the sentences used to represent particular situations. Choice of inference system is by convention; it is clearly a pact between compiler-writer and programmer: that is, between logic-program developer and knowledge representer. There has never been the suspicion in AI that an epistemic situation could be described by an *ideal* set of sentences in an ideal language. Most of the day-to-day research involves freely altering existing axioms and languages. There are those who perceive deduction to be a privileged mode of inference, possessing attractive properties with respect to certain mathematical interpretations; they represent one faction among many.

At one point, I called this situation the "curse of Frege" (openly borrowing the phrase from I. Levi): that there should be an inkling of privilege among the contending factions, while an extremely formal view of representation and reasoning invalidates a priori claims about the superiority of one inference system over another. Of course, this is just the curse of conventionalism, and it might as well be attributed to any number of conventionalists. I held merely that unselfconscious faith in a Fregean kind of inference is what prevents more widespread understanding of the conventionalist view.

Something similar is afoot here. On this formal, conventionalist view of inference and language, no special status can be claimed for deduction, as non-ampliative, as opposed to induction or analogy, which I think should also be called non-ampliative. It would be simple enough to say that induction is ampliative *relative to deduction*, while deduction is not ampliative relative to deduction.[3] This would be separation by fiat.

There is a much better distinction to be made along the ampliative/non-ampliative lines. But it is a distinction that can perhaps not be understood without explaining a wholly different kind of inference, which is based on dialectic. It certainly cannot be understood without appealing to computation.

There are three theses here. The first is that non-computationally conceived inference merely expands notation. This includes induction as well as deduction, and thus both deserve the adjective non-ampliative. Deriving entailments merely expands shorthand. All of the familiar formalisms for reasoning do just this. There now exist examples of formalisms for reasoning that do something else. They are deliberative, and to say in what way they are deliberative requires reference to the process through which they compute their entailments. I explain this, my second thesis, subsequently. Perhaps I should call the familiar

[3] a thought I owe to P. Hayes.

formalisms *rewriting*, as opposed to the latter, which are *dewriting*.[4] In this way, I could avoid debating the use of the term ampliative. But I claim, as my third thesis, that the original ampliative/non-ampliative terminology best survives as referring to this new distinction. So I retain the older terms. Viewed formally, all other attempted distinctions either presume deduction to be privileged, or else fail to separate inference that actually tells us something new from inference that simply rehashes what has already been represented.

2

The first thing to note is that induction, once formalized, expands shorthand as shamefully as deduction. The shame is clear for deduction. Clearly, writing "a" and writing "b" commits one to writing "a & b": that is, when the writing is done in L_D, a language with meaning postulates that encode patterns of deductive inference. In the case of induction, consider L_I, a language with meaning postulates that encode inductive patterns of inference. Write that one hundred ravens have been black, none have been white, and this raven is a random raven. In L_I, this would entail that the probability that the raven is black is high: in fact, sufficiently high for acceptance of the inductive inference that it is black.

This inference is supposed to go beyond what was represented. It surely goes beyond what could have been inferred from "those same sentences" had they been written in L_D. But it is not clear that the inference was not already contained in the premises. In fact, it was.

In writing that this raven is a random raven, in L_I, having already written down the sampling information, one might as well have written that the raven was black. Blackness is entailed of this raven in L_I. Part of what it means to be random in L_I is that frequencies are inherited as probabilities. We know this is entailed because asserting now that this raven is not black would force a revision; some premise must go: either the tally of the sampling, or the claim that the raven was a random raven with respect to color.

The situation is analogous to writing "b" in L_D after writing "a". Writing "b" in the presence of "a" commits the implicit assertion of "a & b". That is what "b" means in the presence of "a". If the epistemics of the situation were so as not to warrant asserting "a & b", then do not assert "b" in the presence of "a". Similarly, if times and places are such that blackness of this raven is not plain to the eye, then do not assert that this raven is random in the class of ravens. Assert something else: that it *seems* to be random, which does not conspire with the sampling information to entail blackness. Sometimes a language will not express subtleties of epistemic situations upon which we happen: we cannot

[4] "dewriting", as in "de-adjectival", "derived from an adjective," or "decompound," "to compound further." I owe this to Theresa, an English Ph.D. student (who by no means condones my invention of words).

assert "a" and assert "b" without asserting "a & b", in L_D. L_D may not be the perfect language. But it is clear enough that in those situations in which each "a" and "b" are asserted, and those sentences in the language are used to depict the situation, then he who so uses the language thus depicts a situation in which "a & b" is asserted.

This unhappy observation of implicit commitment is the result of asserting explicitly that the raven is random. There are inductive languages in which the implicit commitment is more roundabout.

In L_{I2}, assert just that one hundred ravens have been black and that this raven is a raven. If this is all that is asserted, L_{I2} sanctions the inference that the raven is black (L_I could be I. Levi's language, and L_{I2} could be that of H. Kyburg). This appears ampliative; it seems to go beyond the premises. Asserting now that the raven is white, for instance, would require no retraction. In fact, whiteness would be entailed by further adding that the raven is albino.

I will treat this case by explaining first the non-ampliative character of an AI non-monotonic language. By analogy, L_{I2} will be seen to be also non-ampliative.

Consider Bob Moore's auto-epistemic non-monotonic language, L_{AE}. Write in L_{AE} that if I don't know I have a brother, then I don't have a brother. That is all that is asserted. In particular, do not write that I know I have a brother, nor that I know I don't have a brother. Apparently, I just haven't reflected on whether I have a brother. These sentences are supposed to represent this state of affairs. In fact, they do not. In L_{AE}, this sentence alone entails that I don't have a brother (see Moore for details). The description of the world by

"$\neg KB \supset \neg B$"

in L_{AE} entails "$\neg B$" as part of the description. So saying that the world is such that

"$\neg KB \supset \neg B$"

in L_{AE} is tantamount to saying the world is equally well described by

"$(\neg KB \supset \neg B)$ & $\neg B$".

The sentence, "$\neg KB \supset \neg B$" in L_{S5}, a modal epistemic deductive language, does not entail "$\neg B$". So it would seem that "$\neg B$" is an inference of sorts, ampliative with respect to L_{S5}. But that is a ruse. Simply, L_{AE} is selectively inexpressive. There is no way to express just

"$\neg KB \supset \neg B$"

in L_{AE} without also expressing "$\neg B$". Similarly, there is no way to express

$\vdash$ "a" and $\vdash$ "b"

in L_D without also expressing $\vdash$ "a & b".

Let me lampoon the situation to expose it starkly. Suppose a computer has a 32-bit word that is supposed to represent an integer. Ostensibly, it will represent 2^{32} integers. But imagine that there is no way to load a 32-bit word without enduring a computation that reduces the integer to its parity:

"00000...000" for even parity strings, and
"00000...001" for odd parity strings.

It may seem that this computer can represent 2^{32} integers: loading

"00000...110"

would represent the integer six (6) in a normal computer. But in fact, its expressiveness is limited to two states: all other strings of 0's and 1's are disingenuous shorthands. Users of the machine beware! Likewise it is with languages and their associated inference rules. Construing inference as an association between a set of sentences and the set of sentences that is their entailment simply reduces the range of language with which to represent knowledge.

There is a further point about non-monotonic languages which is customary to make. Adding "B" to "$\neg KB \supset \neg B$" does not invalidate a premise. The description

"$B \ \& \ (\neg KB \supset \neg B)$"

is a useable description in this language, unlike

"$a \ \& \ b \ \& \ \neg(a \ \& \ b)$"

in L_D. But this does not diminish "$\neg B$" in any way as an entailment of (just) "$\neg KB \supset \neg B$" in L_{AE}. It simply means that the useable description

"$B \ \& \ (\neg KB \supset \neg B)$",

one way the world could be described, can be formed by *adding* symbols without *subtracting*, starting with

"$\neg KB \supset \neg B$",

some other useable description in this language.

But this is not significant. The string "00000...101" can be had from "00000...100" by turning on bits without zeroing any. It remains the case that "$\neg B$" follows from just "$\neg KB \supset \neg B$" in as much as it remains the case that "00000...1000" has odd parity; hence, our strange computer reduces it to "00000...001." It is a property of the syntax: an accident of patterns of marks. Some shorthands are had by removing symbols. Some are had by removing some and adding others. Non-montonicity of this kind merely defines a different kind of shorthand.

In AI, everyone had thought that this static, syntactic non-monotonicity had something to do with revision of belief. Evidence mounts as time passes, apparently forcing sentences to be added to the previous description of the world. But the meaning of sentences in languages is as intimately tied to what is *not* in the description as to what is in the description. This has yet to be fully appreciated. So adding sentences representing new observation in these languages is as much a revision of the original apprehension of the situation as outright change: deleting and then adding.

What allowed "$\neg B$" in L_{AE} from "$\neg KB \supset \neg B$" was the fact that "$\neg KB \supset \neg B$" was *all* that was asserted. Technically, it is an artifact of the metalinguistic definition of the inference relation, which has as much access to the relation "Not-Asserted", as to the relation "Asserted" (if that makes no sense, take a quick look at how auto-epistemic and other non-monotonic reasoning is defined).

Now return to the hundred ravens. This raven may not be random; it may or may not be albino. But that fact that *nothing has been said* about the raven that might interfere with its being random is significant in L_{I2}. Saying nothing about lack of randomness, saying nothing about properties that might interfere with the direct inference from the largely black class of ravens, is shorthand for saying that it *is* random. This is the nature of L_{I2}.

How can one say that one hundred ravens have been black, in L_{I2}, without saying that a raven about which nothing else is known, is in fact black? Perhaps it cannot be done, like trying to represent six in our strange computer. That is a limitation of the language, with which its users must cope.

Saying only that the hundred ravens have been black, and saying no more, in L_{I2}, amounts to saying that the raven is black. To call the inductive inference that it is black ampliative is a misnomer. It amplifies the premises in a weak sense: if these exact sentences had been transliterated in L_D, symbol for symbol, they would still be well-formed (an accident of syntax), and blackness would not have been implicit. But clearly, using L_{I2} requires more dexterity than that.

One cannot use L_{I2} thinking that one is using L_D. The same sentences in L_D represent quite a different state of affairs. Translations from L_D to L_{I2} and back must be more sophisticated: they must pay closer attention to the entailments under each language. Translation is more than transliteration.

Perhaps inductive inference in L_{I2} is an ex-post policy applied to sentences written by users of L_D. The sentences "One hundred ravens have been black" and "This is a raven" are sentences in L_D, implying what they do in L_D, not what they would in L_{I2}. Once written, as if in L_D, the meaning postulates of L_{I2} are then applied. In this fashion, L_{I2}'s inductive patterns of inference amplify what was committed in L_D. A good analogy of this among non-monotonic languages is circumscription. In circumscription, an assumption is made *after* sentences have been written in L_D: a predicate's extension is assumed to be limited to that which it is asserted to predicate.

This is self-deception, not inference. It is L_{I2} that is being used, or $L_{Circumscr}$, but not L_D. Maybe language users are so constituted that the best way for them to use L_{I2} is to pretend to be using L_D. The thought is perverse, but not unembraceable. It might be that for a certain community of programmers, the most efficient and error-free C++ programs result from telling the programmers that they are programming in C. In any case, pretending to be representing knowledge in L_D while actually representing in L_{I2} is still a use of L_{I2}. What is interesting is the reinterpretation of ostensible L_D-sentences as L_{I2}-sentences, *not* the rewriting of these sentences to produce their "inductive" entailments.

This reinterpretation may be interesting, but is not a candidate for ampliative inference. There is no formal difference between reinterpreting sentences in some language as L_{I2}, and representing the knowledge in L_{I2} in the first place.

3

Beware that sympathy for this view of inference, representation, and language produces a dilemma. As long as L_X is formal, for any X which is supposed to encode some interesting form of inference, computing entailments in L_X seems non-ampliative. It does not venture beyond the premises in an interesting epistemological way. All of the interesting epistemology occurs when a language's "admissible states" (J. Doyle's lovely phrase) are matched with apprehensions of world-situations.

Computing entailments can be surprising for resource-bounded or error-prone users of the language. The conclusions of long proofs are contained wholly in their premises, yet are not obvious. I do not want to suggest that inference in these languages is trivial. But neither surprise nor opacity makes inference ampliative, lest some deductive inferences qualify as ampliative. If deduction, as well as induction, should be called ampliative because there can be long, surprising proofs in both, or undedidable truths, that is a fine use of "ampliative." What I am disputing is the claim that there is an interesting sense in which induction is ampliative, while deduction is not. No deductions are ampliative. So induction, and other alleged ampliative inference, must not be ampliative either. At root of this problem is that all non-deductive patterns of inference, to date, have been formalized in (meta-)languages that encode deductive patterns of inference.

What, then, is ampliative about any *formal* system of inference?

Jon Doyle has identified a simple example of what would seem to qualify as inference going beyond what has been represented in a language: as genuine ampliative inference. Giving his example steals the thunder from a different example that I find more worthy. And Doyle's putative example is of dubious rationality; it may be ampliative inference, but maybe not *rational* ampliative inference. Nevertheless, the example shows immediately what would be required to escape the rewriting nature of all the inference considered above: to differ from all the inference systems defined in terms of entailments.

Doyle cites the example of credulous non-monotonic reasoners. These are the non-monotonic languages that make a non-deterministic choice among several potential conclusions. There are situations in which the sentences in a non-monotonic language can have multiple extensions, or multiple fixed points. Whereas deductive closure is a function from sets of sentences to sets of sentences, non-monotonic closure becomes a relation here.

Writing in L_{AE} just

$$\text{"}(\neg KB \supset \neg B) \;\&\; (\neg K\neg B \supset B)\text{"} \qquad (1)$$

is ambiguous between

$$\text{“}(\neg KB \supset \neg B) \ \& \ (\neg K\neg B \supset B) \ \& \ \text{B”} \qquad (1a)$$

and

$$\text{“}(\neg KB \supset \neg B) \ \& \ (\neg K\neg B \supset B) \ \& \ \neg\text{B”} \qquad (1b).$$

This is different from the sense in which "a" in L_D is ambiguous between "a & b" and "a & $\neg b$". "a" in L_D is neither one nor the other. But (1) is supposed to be either the same as (1a) or else the same as (1b). On some definitions of L_{AE}, (1) describes an admissible state; it is a meaningful, useable description of a state of the world which is different from (1a) and also from (1b). On other definitions, it is inadmissible. Both kinds of definitions are by now familiar cases: in the former, the expressiveness of the language is augmented; in the latter, a shorthand has been disallowed. On the third, more interesting kinds of definitions of L_{AE}, (1) is defined to be a shorthand for (1a) or for (1b), but no one could tell which, prior to the non-deterministic computation.

That is, prior to inference – ampliative inference – it is not determined which state is admitted. The representer of knowledge is responsible for limiting the choice, but cannot be held responsible for the ultimate outcome. The inference "B" or "$\neg B$" is new; it amplifies what was originally represented. Of course, it is a lousy inference: no more than a guess.

So some kinds of bad inference are ampliative. Is any good inference ampliative?

Doyle thinks that certain decision-theoretic deliberations are ampliative. If the underlying decision-making is dialectical in the sense that I elaborate next, then I agree. If it merely expands represented preferences, implicit in an explicit utility function, in some decision-theoretic L_{DT}, then I believe they are merely rewriting, hence not examples of ampliative inference.

4

Something I call dialectic produces rational, ampliative inference: it is dewriting. Dialectic is based on new ideas in the formalization of personal, deliberative inference, and I believe it is the candidate most deserving to be called ampliative. Dialectic makes sense only for defeasible reasoning, and moreover, only for inference defined relative to a set of constructed arguments that may not exhaust all constructible arguments. That is, dialectical inference produces different conclusions, as more arguments are constructed.

Unlike L_{AE} and other first-generation non-monotonic reasoning, defeasible reasoning is often based on the production of arguments. Arguments combine reasons to chain from premises to putative conclusions. They are like proofs, except that proofs (in L_{AE}, L_{I2}, or L_D) establish their conclusions once and for all. A proof is a proof, irrespective of what other proofs there may or may not be. In contrast, arguments justify their conclusions if there are no effective counterarguments or rebuttals. An argument can defeat another argument; an argument that defeats an argument that counters a third argument can reinstate that third argument. Formalisms for this kind of reasoning have been developed by recent authors (Poole, Loui, Pollock, Simari, etc.).

Generally, resource-bounded defeasible reasoners are governed by a process that constructs arguments over time and determines what is justified relative to what arguments have been constructed. Entailment in L_D could likewise have been defined over time, relative to increasing sets of constructed proofs. This having been done, the difference would be that entailments grow monotonically in computation for L_D. For defeasible reasoners, sentences justified at a time bear no such relation to sentences justified at a later time. Conclusions are nonmonotonic in computation. This is what essentially frees them to reach beyond explicit and implicit representation.

Dialectic refers to a set of policies for constructing arguments, that is, permissible search strategies for deliberation. Given what is currently justified, dialectic mandates that attention be focused on constructing arguments for some propositions and not others. Dialectic defines one style of deliberation for resource-bounded defeasible reasoners.

Consider, for example, resource-bounded defeasible reasoning about action. Here is the qualitative case, which belies the merit of the quantitative case, but which is more perspicuous. The explicit commitment is that

> driving quickly results in arriving sooner;
> driving quickly results in endangering lives;
> driving quickly results in having fun;
> and so forth;
> arriving sooner is desirable;
> endangering lives is undesirable;
> having fun is desirable;
> (having fun)+(endangering lives) is undesirable;
> and so forth;
> if a results in p and a results in q then a results in $p + q$;
> a resulting in p is reason for a being as desirable as p;
> do a if a is desirable;
> do not do a if a is desirable.

In a small amount of computation time, there may only be time to identify the argument to drive quickly based on the fact that driving quickly results in having fun. Since there are no other arguments that can be considered in this time (in particular, no counterarguments), driving quickly is warranted. In slightly more time, an effective counterargument might be constructed, which urges not to drive quickly because it results in endangering lives. At this time, neither driving quickly nor not driving quickly is warranted; the arguments interfere. Still later, the argument not to drive quickly might be constructed, based on the undesirability of the combined result, having fun and endangering lives. This defeats the less specific argument based solely on having fun. At this point, not driving quickly is warranted. And the process continues.

Distinguish between three kinds of non-monotonicity. There is syntactic non-monotonicity, discussed above, in which a notation does not always relate a growth of explicit knowledge with a growth of their entailments. It is a property of a representational language. Further, there are two varieties of temporal credal non-monotonicity.[5] Both are properties of actual beliefs of an agent situated in time, as represented in a language. The first is *external* temporal credal non-monotonicity, when a belief is removed due to a revision forced by observation.[6] This includes simple contractions of the corpus of beliefs and other epistemic shifts normally studied in belief revision. External non-monotonicity is not particularly interesting because shifts of belief normally require contraction prior to expansion, and are therefore non-monotonic. Finally, there is internal temporal credal non-monotonicity, when a belief is removed due to deliberation on the entailments of explicit commitment. Non-monotonicity in computation is internal temporal credal non-monotonicity.

The point is that commitments come and go as the search for arguments becomes more complete. In finite systems, there may be a finite set of arguments, and it is meaningful to think of iterative approximation of the ideal state of deliberation. In the final iteration, the arguments constructed exhaust the space of constructible arguments, and warrant for conclusions is based thereupon. In these cases, under the classical view of commitment, there is a final set of implicit commitments for a set of explicit commitments. The agent committed to these explicit sentences is at all times committed to these implicit ideal commitments. Through computation over time, the agent is attempting to acknowledge just those commitments. The view here is different. First it should be noted that there are not necessarily final commitments in infinite systems. Even in the finite systems, however, the nature of commitment is different. Specifying that sentences are defeasible reasons for other sentences is a commitment to their being used in resource-bounded defeasible reasoning that produces conclusions over time, which may be non-monotonic in computation. It is not just another style of specifying ideal commitment. It is a commitment to all of the intermediate epistemic positions, devolved enroute to the ideal.

If facile users of the language were aware of the exact search strategy being used for the construction of arguments, that is, if this detailed control were a part of the language, then a user would be specifying a function from computation time to a set of entailments. Instead of one set of entailments per set of explicit sentences, there would be a set of entailments per computation step per set of explicit sentences. If this were the case, then inference might not

[5] The terminology is taken from Levi's "temporal credal conditionalization."

[6] It was Doyle who showed me that outright contraction deserved to be considered a non-monotonic move.

seem ampliative. At a time, arguments would be constructed as envisioned, and those conclusions warranted with respect to those arguments would be drawn. Those conclusions would be a rewriting, for that time, of the sentences originally inscribed.

Time is an external variable which serves to index a particular rewriting of the sentences. We could argue at length whether this indexing alone results in inference that is *ampliative.*

Fortunately, there is another source of indeterminacy. The search strategy for arguments may have non-deterministic choices. Those choices are constrained, so that any particular sequence of choices instantiates rational search.

By hypothesis, in resource-bounded defeasible reasoning, dialectical search strategies are rational. Dialectical search implies that resources are at all times being expended to overturn currently warranted conclusions, or else to establish new ones. In the example of practical reasoning above, a dialectical reasoner could not have formed another argument to drive quickly, based on arriving sooner, prior to constructing the second argument, an argument for not driving quickly. Resources cannot be used to buttress arguments further, if there is no existing effective counterargument. Resources could be expended in that way in search strategies that are lobbying, *i.e.*, not dialectical. But to expend resources in that way does not guarantee the rationality of what results at every intermediate stage in the compuation. Perhaps at a time, resources that could have been used to rebut a putative conclusion were used frivolously. All dialectical search strategies are rational, while not all lobbying search strategies are rational.

In writing sentences in a language of resource-bounded defeasible reasoning, the user commits to the results of any dialectical strategy, but does not know in advance the exact sequence of dialectical maneuvers. For example, an argument that could be challenged at any of a number of places will be challenged, though it is undetermined where the challenge will be made. The difference between this arbitrary resolution of indeterminacy and the resolution of multiple extensions by arbitrary choice of extension has to do with the claims of rationality of the process. Non-deterministic choice of maneuver is rational; non-deterministic choice of conclusion is not.

Dialectic produces more than just sentences that rewrite the commitment, but nevertheless, produces only sentences to which the agent has contracted to commit. It is as if there were reason to commit to the outcome of a coin toss between p and $\neg p$. The coin is tossed: the inference made.

What we have been seeking is a language in which a prior commitment to the outcome of an indeterminate process can be specified. The process must itself be rational (this excludes guessing). And the indeterminacy cannot merely be the time at which the inevitable outcome is reached (this excludes non-deterministic use of deductive inference patterns). That is how inference can be more than the rewriting of commitment, but more than a stab in the dark. That is what constitutes ampliative inference.

Ampliative inference is the result of rational non-deterministic non-monotonic computation.

Dialectic is the very best example of rational ampliative inference. We saw that non-deterministic choice of non-monotonic extension is also an example. There are others. Abduction vaguely described as the inference to some causally sufficient condition might be ampliative. It has affinities to the multiple extension problem; there is an indeterminate specification of conclusion and a choice to be made with no formal guidance. Inference in connectionist networks is ampliative. There is a computational process that produces tentative conclusions through time. The time at which the network is inspected for its conclusions is often externally determined. Conclusions produced at one time may be overturned at a later time. There may be other approaches to ampliative defeasible reasoning, too.[7]

5

Non-monotonic logics, by their form, suggest how internal credal non-monotonicity might be achieved. But unless there is concern over resource-bounded computation, they exhibit only syntactic non-monotonicity, which everyone agrees is a matter of style.

Traditional logicians[8] deny interest in non-monotonic logics because they disavow computation; they care only about the specification of an agent's commitment. What we find here is that the reverse is true. In order to conceive of an inference that is not merely a rewriting of the agent's specified commitment, we have to turn our thoughts to computation.

6 References

Doyle, J. "Constructive belief and rational representation," *Computational Intelligence 5*, 1989.

Elgot-Drapkin, J. and Perlis, D. "Reasoning situated in time," submitted to *Artificial Intelligence*, 1989.

Geffner, H. and Pearl, J. "A framework for reasoning with defaults," in H. Kyburg, R. Loui, and G. Carlson, eds., *Knowledge Representation and Defeasible Reasoning*, Kluwer, 1990.

Ginsberg, M. *Readings in Non-Monotonic Reasoning*, Morgan Kaufman, 1988.

Kyburg, H. *Logical Foundations of Statistical Inference*, Reidel, 1974.

[7] Consider non-deterministic versions of step-logics, *e.g.*, Elgot-Drapkin and Perlis, or resource-bounded versions of irrelevance-introducing systems, *e.g.*, Geffner and Pearl.

[8] This comes from Isaac Levi, in particular.

Levi, I. *Enterprise of Knowledge*, MIT Press, 1980.

Loui, R. "Defeat among arguments," *Computational Intelligence 3*, 1987.

Loui, R. "The curse of Frege," *Proc. Theoretical Aspects of Reasoning about Knowledge*, Morgan Kaufman, 1988.

Moore, R. "Semantical considerations on nonmonotonic logic," *Artificial Intelligence 25*, 1985.

Poole, D. "On the comparison of theories," *Proc. Intl. Joint Conf. AI*, Morgan Kaufman, 1985.

Pollock, J. "Defeasible reasoning," *Cognitive Science 11*, 1987.

Sampson, J. and Weiner, E., preparers. *Oxford English Dictionary*, Clarendon Press, 1989.

Simari, G. "On the logic of defeasible reasoning," to be submitted to *Computational Intelligence*, 1989. (Washington U. CS Tech. Report).