Let me say first that I remain as enthusiastic as ever regarding the eventual role in AI of commonsense reasoning in general and nonmonotonic reasoning in particular – both of these areas seem to have become accepted as central to the eventual success of our discipline.

I have therefore chosen to focus on what strike me as serious problems with current nonmonotonic work; briefly put, I believe that we have put ourselves in a position where it is almost impossible for our work to be validated by anyone other than a member of our small and ingrown subcommunity of AI.

The reason that this is so important is that validation is what distinguishes science from alchemy. The fundamental goal of constructing an intelligent artifact (or turning lead into gold) is a grandiose and difficult engineering task. Enthusiastic pursuit of this ambition is fine, but if we wish to lay claim to a *scientific* pursuit, we must identify and solve subproblems that are recognizably on the critical path between us and our goal.

The early work on the formalization of commonsense reasoning was motivated by concerns such as these. More recently, however, we have lost track of our fundamental objectives. Instead of focussing on the result of applying our intuitions to a broad range of problems, we have focussed on a few trivially simple examples. Instead of remembering that we are an engineering discipline and concentrating on implementations, we have ignored our roots and concentrated solely on mathematics. (Consider the current fascination with model-theoretic approaches to nonmonotonic inference.)

I am especially troubled by what seems to me to be a trend not to formalize our intuitions, but to get the "right" answers to a small set of simple problems. What reason is there to believe that a formal system designed in this fashion will be of use to the AI community as a whole? I am aware, of course, that authors generally believe that they *have* formalized their intuitions – this is why it always strikes me as surprising when there is a wealth of counterexamples to their suggestions. How many people working on describing action have actually attempted to implement a planner using their ideas?

The issue here is one of validation. To overcome these difficulties, I would suggest that simulated domains be constructed, and simulated agents should attempt to operate in those domains.[1] These simulated agents may fare well, and they may fare poorly; either way, we will have learned something more significant than their ability to cope with a now-tired suite of simplistic examples. An effective alternative would be for us to tackle reasoning problems of interest to the industrial community.

To my mind, the nonmonotonic community needs this ability to test its hypotheses in a rich environment far more than it needs more progress on theoretical issues. But for those who insist on continuing to work in the paradigm we have developed, I would suggest the following topics:

First, we need to understand causality. Philosophers have been wrestling with this for some time (generally without success), but there is some reason to believe that we are more likely than they to make progress – we have identified some specific areas in which tractable problems appear to depend on causal notions for their solution. The analysis of inheritance hierarchies is one such area; the formal methods that have been presented here are struggling to describe conditions under which the success of one default causes the defeat of another. Reasoning about action (especially coming to grips with the qualification problem) is another area where a causal understanding of the domain is crucial.

Secondly, we need to effectively justify the eventual role of nonmonotonic reasoning within AI itself. Nonmonotonic methods were initially proposed as a way to *speed* inference by allowing a reasoning system to jump to default conclusions, but all of our formal work is computationally intractable. We need to find and understand situations in which some form of nonmonotonicity *does* lead to a computational speedup. Can default rules, for example, somehow be used to focus the search of a conventional theorem prover? Here is an application where one can imagine that a failed consistency check might not matter as much as it does in some other domains; if nonmonotonic representation techniques could be used to give a declarative formalization of heuristic control rules, that would be significant progress.

[1] *Real* domains would obviously be better, but operation in actual domains is beyond the practical abilities of existing robotic agents. Note also that we are assuming that the designers of the agents and of the environment are different; this will *hopefully* prevent the agents from succeeding by exploiting loopholes in the simulation.