# Epistemic virtues, metavirtues, and computational complexity

Adam Morton

to appear in *Nous*. This is the version that will appear in *Nous* plus two appendices, and with some navigational links. (Note on hyperlinks in pdf – they take you to the page in question, but not always to the exact point on the page. Sometimes you have to scroll down to find the target.) This version for cogprints was prepared in April 2003.

Abstract: I argue that considerations about computational complexity show that all finite agents need characteristics like those that have been called epistemic virtues. The necessity of these virtues follows in part from the nonexistence of shortcuts, or efficient ways of finding shortcuts, to cognitively expensive routines. It follows that agents must possess the capacities – metavirtues –of developing in advance the cognitive virtues they will need when time and memory are at a premium.
keywords: **cognition**, **computational complexity**, **epistemology**, **epistemic virtue**, **metavirtue**, **virtue**.

# Epistemic virtues, metavirtues, and computational complexity

In der Beschraenkung zeigt sich erst der Meister. Goethe

A master is a craftsman who knows how to begin, how to continue, and how to end. WalterSickert (about Whistler)

Do all your work as though you had a thousand years to live, and as you would if you knew you must die tomorrow. Mother Ann Lee, founder of Shakerism

**1. The metaresource catch** You are trying to learn the names of the students in a class. You find that you can recite most of the names reliably, but that in class they evade you. In some contexts they do, that is, but not in others, and you have some idea which. Your first reaction to this is to think as you begin a classroom interaction whether it is one in which you can rely on the names occurring when you need them. But this doesn't work: you find that when you burden your mind with these considerations contexts in which names would otherwise have been no problem become troublesome.

You have never been quick at mental arithmetic. As a child you discovered a trick for adding up columns of figures in your head. It consists in looking ahead for easy combinations and adding them first. So if you are adding 37, 19, 13, and 11, you first add 37 and 13 to get 50, then add the 19 to get 69, with which the 11 easily combines to get 80. You use this method frequently because it means that you less often have a long pause in your calculations. As an adult, though, you find that you very often get the wrong answer with this method, and when you go back to straightforwardly plowing through the figures from beginning to end it proves not only more accurate but at least as fast.

You deal with a slow internet connection by doing tasks in other windows while

web pages are loading. But you get distracted and spend longer on the other tasks than you should, and you conclude that unless the pages load at an uniformly very slow rate you get more done if you do not zip back and forth but just wait it out.

In all of these examples there is a catch to a strategy that seems at first to be a good response to the finiteness of a person's cognitive resources. The catch is that following the strategy may add to the demands on those resources. Call it the metaresource catch. It is a very widespread phenomenon, and various forms of it will recur throughout this paper. The metaresource catch suggests that giving advice for managing cognitive limitations is a very subtle business: even if we manage to describe a pattern of thinking which would avoid overload of our capacities to remember and manage complexity we may find that understanding and following that pattern may add to rather than reduce the overload. {note 1}

The aim of this paper is to work out a particular form of this suggestion in a way that makes a connection between the theory of computational complexity and the concept of an epistemic virtue. The general strategy is best seen by distinguishing between virtues that derive from our emotional nature and those which derive from our intellectual limitations. We need moral virtues such as courage, and its epistemic counterpart of intellectual adventurousness, largely because of our susceptibility to specific emotions, e.g. timidity. In principle we could formulate a rule that would say when the courageous agent would stand up to aggression, or investigate an unpopular hypothesis, and then an agent who was not subject to fear could simply follow the prescription {note 2}. (I am not saying that we have any such prescription, just that it would make sense for a philosopher to try to formulate it.) But since we humans are subject to fear, we need the virtue of courage, to find and follow the right path when part of our nature is pulling in the other direction. It is tempting to think that virtues of limitation management – I give examples below – are like that: we can in principle say what the best way to manage the situation is, and this would be advice that if only we were not subject to the limitations we could simply follow to good effect. But virtues of limitation management – or more carefully the limitation-management aspects to most

virtues – are not like that. Just how unlike that the paper tries to bring out.

My way of developing my theme will be to make connections with an existing literature that wrestles with very similar problems. This is the theory of computational complexity – the theory of the different degrees of tractability of computational problems. I argue that if we articulate the problems of limited cognitive capacity along the lines suggested by this theory then we find states of mind with the general characteristics of epistemic virtues inevitably appearing. Their appearance is directly related to cognitive/computational limitations. The demonstration of this has another consequence, besides the theme of the distinctiveness of limited rationality: the existence of epistemic virtues for cognitively limited agents can be proved. To begin, though, I must state what the general characteristics of epistemic virtues are.

**2. A framework for virtues.** Epistemic virtues are characteristics of people that help their knowledge-acquiring projects succeed. (Just as moral virtues are characteristics that help people's projects of living well, individually and together, succeed.) Not any such characteristic is an epistemic virtue. For example intelligence does not count, nor sanity, nor knowing the meaning of Arabic words, though all of these are knowledge-conducive characteristics. Standard examples of epistemic virtues are care with evidence, resistance to the urge to form hasty conclusions, and imagination with respect to explanatory hypotheses. Controversial or borderline candidates for being epistemic virtues, for various reasons, are a good memory, sociability, and the capacity to acquire languages. It is important to see that epistemic virtues are not simply components of a single capacity for rational belief formation. For the current debate on the topic raises hard questions about the relations between belief-forming capacities. Some virtues can oppose one another – for example caution and fertility - and the same characteristics can help and hinder knowledge, in different people at different times. Contrast for example the traits of mind required by long-range climate forecasting and field botany. (Mathematical modeling and basic empirical science.) Each draws on a range of intellectual capacities, but it is unlikely that someone who had the capacity for simplifying abstraction required for the one would

have the patient accuracy required for the other.

What would epistemic virtues be like if they were to play an essential role in describing how we do and should form our beliefs? The constraints turn out to be quite demanding. I shall take an epistemic virtue to be any intellectual characteristic V of a particular agent which has the following six features {note 3 } :

- *helpfulness:* there are circumstances in which possession of V increases the likelihood of a person achieving a desired epistemic end (such as truth, usefulness or explanatory value.)
- *versatility:* there is a wide variety of such circumstances, varying in the kinds of beliefs acquired.
- *specificity:* there are circumstances in which possession of V does not further an epistemic end
- *non-knowledge:* V does not consist in the possession of particular items of information
- *non-reduncancy:* V can not be replicated by the agent's capacity to carry out correct inference
- *counterfactual senstitivity:* V is counterfactually sensitive to conditions under which it is likely to facilitate a belief-forming process: if conditions had not been suitable V would have been less likely to influence the agent's cognition.

Each of these is a very natural candidate for being a necessary condition for epistemic virtue-hood. (I shall return to the sixth condition below.) It is much less evident that the conjunction of the six provides a sufficient condition. But they provide a good enough starting definition that when in what follows I have shown that some characteristic satisfies them I shall take myself to have made a good case that it is an epistemic virtue. The fifth condition, non-redundancy, appeals to a prior understanding of correct inference. It is correct inference within the particular agent's capacities, though: virtue does not enter when you could have got the same result just by careful thinking. (Just as a moral virtue is a trait that is not redundant given the agent's capacity to understand and reason on the basis of moral principles.) So reasoning power, in the sense of the bare capacity to reason, whether or not in valuable

directions, is ruled out by definition, and memory is ruled out because it is always helpful. Knowing the meanings of many Arabic words is ruled out as possession of evidence, though the capacity to acquire and use languages is not ruled out, though it does seem to have important differences from the virtues that I shall discuss. The best examples of characteristics that fit the profile are what I shall call H and C virtues. (H for Harman and C for Cherniak {note 4}.) H virtues are useful because the fact that beliefs $B_1,\ldots,B_n$ entail a conclusion C, or that the probability o f hypothesis H is high given accepted evidence E, does not always make it a good idea to add C or H to one's stock of beliefs. It may be better to reconsider some of $B_1,\ldots,B_n$ or to rethink one's acceptance of E. It may be better in that one's resulting beliefs may contain fewer falsehoods, or be more intellectually fruitful, or more helpful in some application. Our capacity for inference is not crippled by this fact because we possess, to varying degrees, characteristics with everyday labels such as conservatism, stubbornness, and doggedness. (Contrast the Moorean virtue of saying 'whatever the argument, if it shows this something is wrong' with the Russellian virtue of saying 'if that's where it leads me, that's where I go'.) We can apply these labels because people possess a swarm of capacities to evaluate the plausibility, usefulness, and promise of sets of beliefs, given their overall cognitive and practical situations, and to consider various combinations of acceptance and rejection. These capacities are, for deep and universal reasons that I discuss below, inevitably more demanding of cognitive resources than the inferential processes that occasion them.

C-virtues are useful because any starting point for reasoning leads to a branching maze of consequences or supported hypotheses, most of which are irrelevant or uninteresting to the intellectual or practical point at hand. We avoid becoming lost in the maze because we have capacities that have everyday labels such as caution, foresight, stubbornness, and courage. Again these labels hide a range of cognitive capacities, for evaluating inferential strategies, for producing relevant hypotheses, and for assessing their promise. And again there is a fundamental difference in cognitive expense between these capacities and the inferential processes to which they are related.

Such characteristics are typically variable: different people have them in different degrees. If they are to satisfy the fifth condition of the list above some variability is inevitable. For suppose that there were a clearly discoverable best way to perform some inference-regulating function. (For example a best way of judging which of a group of premises that lead to a contradiction is most likely to be responsible for the trouble.) Then an intelligent agent could simply reason that this was the best way, and put it into practice. Given true beliefs about reasoning power and the ability to perform a routine that one has reason to believe is optimal, other capacities would be redundant. So if there are epistemic virtues, fitting the profile above, inference cannot be self-regulating {note 5}. In section 4 below I state carefully why it cannot be, and therefore why there is room for H and C virtues. No such non-redundant virtue can consist in performing some identifiable operation as much as possible, or avoiding some class of operations, or in general in setting any variable to 'max' or 'min'. For this would make it redundant: one could simply infer the advice to follow the universal prescription. So such virtues will have an Aristotelian quality: exhibiting them will mean finding a hard-to-define mean point between extremes. There will always be room for a variety of defensible settings, though there are clearly indefensible ones. In fact, subtle means are more clearly a feature of epistemic virtues than of moral virtues. Perhaps, just perhaps, one cannot be too tolerant of other people, but one can be too tolerant of daring hypotheses, and also not tolerant enough {note 6}.

Virtues have forward-looking and in-the-moment aspects. Consider the non-redundancy of a virtue such as courage. If standing up to a bully can clearly be seen on general principles as the right thing to do then deciding that that is what you should do does not require courage, though carrying it through, actually making yourself look him in the eye and say "No. Do your worst" may. So in this case the prospective aspect of courage is redundant, given enough intellectual capacity. The distinction here, between the virtue of adopting the courageous course and that of carrying it out, applies to epistemic virtues too. Consider for example the virtue of being able, when appropriate, to invest a specific amount of time and intellectual effort into an inferential strategy that may not give immediately interesting results. This too divides naturally

into two parts: there is the capacity to know that that amount of time and effort is likely to bring results, and there is the capacity actually to execute the strategy to that extent. The sensitivity to the mean resides in the first part: the agent must be attuned to aspects of the situation, including both cognitive and environmental factors, in such a way that she can recognize situations in which a specific degree of commitment to the strategy is appealing. So in many cases the virtue can be split: there are the *prospective* capacities - of getting on a right course - and the *operational* capacities - of staying on it. The prospective aspect gets the ball rolling along a path at some point of which an operational aspect must enter. The operational aspect is not a trivial business, given that the person cannot keep herself to the course by re-deducing its rightness as she goes along. She has to retain the initial resolve, and also be able to sense its further implications for later choices {note 7}.

A prospective capacity pays off when it initiates a line of thinking under the right circumstances. The link with the right circumstances could be entirely accidental. For example someone might always leap to a certain kind of conclusion whatever the evidence. When as it happens the conclusion is true this mental tic would have shown itself to be the prospective aspect of a virtue, on the account so far. This would be like counting as courage the tendency always to take the confrontational option. What we normally call virtues, both epistemic and moral, involve a prospective component that is non-accidentally sensitive to the circumstances under which the operational component will pay off. They embody a kind of special-purpose low-grade knowledge that the circumstances are right. The knowledge is not infallible; sometimes and perhaps often the agent will set intellectual sail in the face of a storm. But when it succeeds it is not simply by chance. That is why I have included the sixth condition, counterfactual sensitivity. This is an essential feature of epistemic virtues, a central part of why limited agents must have them and why we appeal to them for everyday normative and explanatory purposes.

An example to end this section, illustrating the main themes. An experimental psychologist, call her Alice, has a remarkable gift for devising experiments which will test hypotheses about childhood cognitive development. Her designs are natural, so

that child subjects can do what is required with minimal intervention from adult psychologists, and at the same time cunning, in that they focus on the differences between rival hypotheses and exclude trivializing unintended explanations. Alice is much less comfortable with statistics, though, at any rate with the sophisticated non-cookbook analyses that some referees for some journals are beginning to want. Luckily she has a colleague, Bruno, who lives and breathes statistics, cookbook and non-cookbook, parametric and non-parametric, Bayesian and traditional. She consults Bruno briefly while designing an experiment, and then has long sessions with him once the data is in, in order to select and present the figures in a way that will lead to minimal hassle from the experts. Now though Bruno is statistics-obsessed he is not quite as expert as he seems; he does not have mathematical insight or comprehensiveness. In designing one particular experiment Alice needs to forestall a potential objection, which would suggest a very complicated distribution of some cognitive skill among 4 year olds. Bruno suggests that a particular randomizing device will neutralize the objection. In fact it will, but not for the reasons Bruno suggests, which are based on a subtle misunderstanding. Alice rarely follows all of Bruno's explanations anyway, but the device seems sensible to her, so she adopts it.

Alison is exhibiting a typical epistemic virtue here. Or, rather, in the situation she is in the capacity she needs has the typical form of an epistemic virtue. To take Bruno's advice she has to be able to rely on her capacity to carry out a plan whose rationale she has only a partial grasp. That is the prospective side of it. And on the operational side, she must in the course of running and writing up her experiments be able to carry through with this capacity for acting on this limited understanding. She will have to be able to handle twists in the data that she had not anticipated. She may or may not be capable of either of these, and her trust in her capacities may or may not lead to a worthwhile result. In some circumstances her willingness to proceed on imperfect understanding will pay off, and in others it will not. She has no assurance here, at any rate none that she can infer from the available evidence. But if she does succeed and if her success is a matter of virtue rather than chance, her capacities will have to be invoked by some relevant aspect of the situation, in spite of her inability to

make the connection in terms of explicit reasoning.

(Why did I choose a competent experimentalist and an imperfect statistician? Because each capacity is demanding in such a different way. Because the great difficulty of statistics combined with its centrality in inductive reasoning suggests that no one person can master the thinking that would be needed to get perfectly from evidence to confirmed hypothesis. Because reliance on flawed information in which there is a large but not-easily-separated proportion of truth is typical of many scientific situations. For example it often accompanies mathematical modeling. And in such situations the capacities we need are much subtler than those needed to start from some assumptions and see what is probably true if they are.)

**3. Cognitive complexity** Epistemic virtues require a delicate balance of opposites. They have a disorderly and an orderly aspect. The delicacy emerges most clearly with the tension between the non-redundancy condition and the counterfactual sensitivity condition. Virtues are not needed either to manage or to describe situations where inference, deductive or probabilistic, is adequate by itself. But on the other hand they are not found where success is determined by pure chance. Inference includes second order reasoning, so when a virtue is both appropriate and non-redundant it will either be impossible to reason straightforwardly to the conclusion that it is appropriate, or impossible to reason straightforward to a recipe for following it. (You can sometimes know exactly when a virtue is called for, and you can sometimes know exactly what someone exemplifying the virtue would do, but you can rarely know both.)

The delicacy may seem paradoxical, even contradictory. Sometimes the appropriateness of a cognitive process is both known and unknowable, and at other times its content is both available and indescribable. The paradox can be blunted by defining terms more carefully. We can have some kind of knowledge of the appropriateness of a virtue when we cannot have some other kind; we can have some kind of knowledge of the consequences of acting in accordance with a virtue when we cannot have some other kind. The problem is defining these linked but distinct kinds of

knowledge. The connection with cognitive limitations is immediate here. Non-redundancy requires that an agent not be able to derive the appropriateness of a pattern of behavior or the promise of a line of inquiry by direct reasoning, given the constrains upon her. Thinking it out directly isn't a good strategy for her. So it is not-too-expensive knowledge of a virtue's appropriateness that we cannot have. How are we to think of cognitive expense? We don't really know: it is as likely that an enlightening account of epistemic virtues would inform us how to understand cognitive expense as that developing an account of cognitive expense will yield interesting conclusions about epistemic virtues. It seems clear to me that we should apply pressure from both ends. One resource to call on comes from computer science, where there is a well worked out and formidable account of computational expense, complexity theory, the study of the inherent difficulty of problems, measured in terms of the computational resources it takes to solve them.

The theory of computational complexity is based on the fact that some computational processes are inherently more expensive than others, measured in the time taken to complete them, the amount of working memory ('space') they require, or more mundane factors of ink, electricity, or glucose. And most importantly, the degrees of expensiveness fall into definite classes, between which there are systematic relations. It would be very unlikely that the full structure of these classes transfers in any useful way to the domain of cognitive expense of problems for human beings. However, there are some results and some trends of complexity theory that are so general that it is hard to see how they could fail to apply to any cognitive process of any finite creature. As I will show in this section, these results allow us to recreate the intuitive idea of an epistemic virtue. If we take straightforward reasoning to be cognition of a relatively low order of complexity, then more complex but perfectly un-mysterious processes that are related to this cognition in specific ways will exhibit the defining characteristics of epistemic virtues.

The universality of some of the results of the theory of computational complexity can be most easily grasped by starting with their precursors, the great metalogical theorems of the 1930s. One theme running through these results is the greater

difficulty of determining consistency than of carrying out deductions. Goedel's incompleteness theorems together show that though when a sentence is provable in axiomatic arithmetic its provability can be proved in arithmetic, when a set of sentences in the vocabulary of arithmetic is consistent that fact may fail to be provable in arithmetic. Church's theorem, interpreted in terms of recursive functions, shows that though the question whether a sequence of formulas is a deduction in first order logic can be answered recursively, the question whether a formula is *in*consistent (i.e. whether there is a deduction of a contradiction from it) can often only be answered recursively enumerably (in that the set of formulas to which the answer is Yes is r.e.). And the question whether a formula is consistent cannot in general be even recursively enumerably answered. These results are however directed at ideal computation: even the recursive functions include procedures well beyond the reach of any conceivable computer and certainly any human reasoner. Contemporary complexity theory manages to scale these results down to much more down to earth conceptions of what can be computed.

There are two central scaling down devices. The first is a classification of computations into classes according to the rates at which their demands for resources increase with the size of their input. I will not discuss this further {note 8}. The second is a contrast between deterministic and nondeterministic computation. A deterministic computation establishes that a given algorithm will take a specific input to a specific output, while a non-deterministic computation establishes that an algorithm will give a specific output given some input or other. So every algorithm can be used in a deterministic and nondeterministic way. As a result, for every class of problems that can be solved by a class of deterministic computations there is a corresponding class of problems that can be solved by the corresponding nondeterministic computations. The best known such pair is that of P and NP: problems that can be solved in time that increases as a polynomial function of the size of the input and problems that could be solved in polynomial time if some oracle were giving suitable inputs to the computation.

The relation between deterministic and nondeterministic computation generalizes the recursive/recursively enumerable contrast in a way that applies it to more realistic

categories of computation while preserving many of its significant features. In particular the deterministic/nondeterministic contrast to a central cognitive fact, the distinction between inference and consistency. In the simplest case, that of reasoning in propositional logic, the question whether a sequence of formulas is a deduction is answerable in polynomial time, while whether a formula is consistent is answerable in nondeterministic polynomial time (in fact it is as hard as any NP problem can get), and whether a formula is inconsistent is at least NP and possibly harder. So as in the case of quantificational logic questions of consistency are of a greater order of difficulty than questions of inference. And the similarity runs deep, since there are fundamental similarities between nondeterministic computation and recursively enumerability: a set is r.e. when there is a recursive function which given an oracle supplying suitable inputs can determine membership in the set. Reasoning using the full resources of quantificational logic outruns the capacities of both humans and computers, so that when a human or a computer program performs a series of deductions they always fall into some logical sub-system, restricted either in terms of syntactical complexity or in terms of the sizes of the models relevant to questions of validity. When such sub-systems of quantificational logic are studied, it turns out that there too questions of consistency are essentially harder than questions of deduction, and are the nondeterministic correlate of the former {note 9}.

So it is universally the case that consistency questions is more of a challenge than the relatively trivial deduction question. And the reason is not very mysterious: the consistency question forces one to explore the branches of the tree of computations of answers to deducibility questions, and the number of branches of a tree are a rapidly increasing function of its depth. It is important to see that these results are not parochial facts about particular computational devices. They follow from assumptions of such generality that it is hard to see how they could fail to apply to any process which consists of discrete steps each of which takes a finite amount of time and which involves use of working memory which is in finite supply. (Another reason derives from the link between complexity classes and types of formal language. Again we have a connection of great generality that appears to apply to an enormous variety of ways in

which though can be instantiated. But an exposition of this link will not fit into this paper. {see appendix one for some remarks on the issue}.) The gap between deduction and consistency is thus to be taken as something that will occur in computers, humans, and almost anything that thinks.

Consider then the situation of a creature that expands its information about the world by inference. A new belief is acquired, by perception or testimony, and then its consequences have to be absorbed. A cascade of new beliefs results: it is just about impossible to acquire just one belief. (I say to you "Saturn is not the only planet with rings." And you immediately think, "Some planet that is not Saturn has rings", then "If Earth does not have rings, then some planet that is neither Saturn nor Earth has rings.") The cascade may begin with little more than immediate consequences of the newly acquired belief, but soon involves interactions with a variety of pre-existing beliefs. We don't have to assume that the cascade is driven by textbook logical deduction. But it must proceed by processes that, like deduction, are not too expensive. We must be capable of seeing enough of what follows from a new item of information to link it with existing information and to use it to guide actions. (When you see a bear on the trail you don't want to solve major intellectual problems in order to link the discovery with information about the dangers of bears and the best ways of avoiding them.)

So there must be processes that produce beliefs from beliefs in a computationally easy and quick manner. (And as most philosophers and artificial intelligence researchers assume, it would be quite amazing if these processes did not draw largely at least on large subsets of the computationally very inexpensive relation of deductive consequence.) Call these processes collectively acquisition. Almost inevitably, acquisition will lead to beliefs that contradict other beliefs. Or, more carefully put, any acquisition process that is computationally efficient will result in contradictions, since checking for inconsistency is an inherently harder business, so that an efficient acquisition process will embody only a few computationally easy special cases of consistency checking. Eventually, these contradictions will have to be discovered, and appropriate measures taken. So we have two other families of

processes, both of them computationally more expensive than acquisition, call them checking and repair. Checking involves discovering contradictions, and more generally tensions, implausibilities, and evidential tangles, all of which require exploration among the infinitely many paths between existing beliefs. Repair follows from the discovery of contradictions and other problems, and involves adding and removing beliefs so as to alleviate deductive and evidential tensions, and making decisions about which difficulties require immediate attention and which can safely be ignored {note 11}. An intuitively attractive idea, for which there is some psychological evidence, is that checking heuristics include constructing mental models – sketchy, incomplete and very finite – for possible situations that might jointly satisfy bodies of belief {note 12}. Thinking with psychologically manageable models is inherently different from thinking deductively. For while deductive thinking in a limited agent is inevitably incomplete – one can never find all the relevant consequences of an assumption – it is at any rate potentially precise. On the other hand since most models one can manipulate mentally are incomplete specifications of possible situations checking that beliefs hold in a model is at best tentative assurance that they are in fact consistent.

A creature with immense computational power could have fixed and precise routines for checking and repair. If its computational power was so immense that it could handle nondeterministic versions of the computations of more ordinary creatures then it just might have a chance of building checking and repair into acquisition. But real creatures are not like this. To accomplish checking and repair they will need approximations and heuristics, which within their limits of time, working memory, and other resources will catch enough contradictions and fix enough of those that are caught to allow the creatures to survive and, if they are scientific creatures, to accumulate true and useful beliefs. In other words, real creatures will need the C and H virtues introduced in the previous section.

There are many procedures and strategies that can be employed within resource constraints, for checking and repair: each of them will leave some potentially important possibilities unexplored, and each of them will interfere to some extent with the unrestrained derivation of new beliefs. A very crude hold on the variety of procedures

can be got by thinking in terms of the relative speed of acquisition and checking. We can slow down acquisition so as to stay in touch with a relatively more thorough checking procedure, or we can employ a fast and inaccurate checking, or we can stick with both fast acquisition and slow careful checking and accept the inevitable discrepancy, switching to a different repair mode when contradictions that cannot be ignored emerge. Different such combinations will be appropriate under different circumstances, and any such combination will have bad consequences in some circumstances. At some times it will be best to rush ahead with deduction and let consistency wait. At others that would be disastrous, and it will be best to proceed step by step, retaining Cartesian certainty at each point. And very often it will be best to follow some course between these extremes {note 13}.

No easy routine, nothing on the order of deduction, will determine which acquisition/checking balance is called for. For the major factor such a routine would have to take account of would be the time or memory demands of checking consistency among the beliefs at hand or likely to be acquired, and there is no easy way of estimating this in advance. In fact there is no way of estimating in advance what beliefs will be acquired in a course of reasoning, since "A is deducible from B", for fixed B, is to be classed with "is inconsistent" rather than with "is a deduction of A". So routines which judge the right balance of acquisition, checking, and repair will be of a high order of difficulty.

We have now seen the need for creatures whose powers of inference fall under definite constraints to employ cognitive routines that deal with tasks that exceed those constraints. (I postpone until the next and final section a discussion of how such routines can become available.) These routines pay off in some situations, and not in others. They do not consist simply in the possession of information (since then they would fall into the category of deduction.) They are non-redundant, since their functions cannot be replicated by inference. And they are sensitive to the situations under which they are called for. So, they are epistemic virtues. Real agents will not employ crude balance-setting routines arbitrating between acquisition, checking, and repair, but rather much subtler and more specific capacities, tuned to the general kinds

of epistemic situations in which they have evolved and the more specific circumstances in which they have learned to operate. But the point is the same. There are epistemic virtues. And salient among them there are processes that manage our cognitive limitations.

**4. few shortcuts** The distinction between inference and virtue might seem fairly superficial. For the deep divisions are between easier and harder computational problems, but a thinking agent that can only manage fairly easy computations will have to use manageable approximations for harder problems, so that acquisition, checking, and repair will in the end all use affordable cognitive routines. The distinction would then just be between more and less approximate processes. And in real cognition even simple inference will usually involve some degree of heuristics and approximation.

The distinction goes deeper than this line of thought would suggest. It is related to a form of the fundamental epistemological division between internalist and externalist modes of justification. To make the link, consider the following question. Why cannot a person integrate her checking and repair into her acquisition, via higher-order reasoning about optimal compromises between accuracy and computational expense? The person would reason from facts about cognition - both apriori facts about computation and logic and what contingent facts about her memory limitations and the like- to beliefs about the strategies for balancing acquisition, checking, and repair that she would benefit from using. The strategies would not have to be perfect or precise: they could incorporate approximations and fallible short cuts, as long as these were well thought out, justified, approximations and short cuts. If a person could do this then she could calculate the best route to take through the various stages of a project, given the constraints on time and cognitive resources, and calculate her likelihood of sticking to the route. She could thus substitute general-purpose rationality for combinations of specific virtues. But there are systematic reasons why we cannot count on being able to find such second-order reasoning. They amount to versions of the metaresource trap of section 1. The thinking that leads to conclusions about what can be done within resource constraints may itself be too expensive. Even if general

principles about efficiency can be deduced from available beliefs, finding the right application of those principles given a description of the situation will require a computation, and that computation may be very hard. (Compare: we can deduce the general equations of fluid flow, but the derivation of useful predictions from them, for meteorology or aerodynamics, is most often wildly beyond reasonable computing possibility.) So there is at the very least a worry hanging over the suggestion: we may not be able to afford effective use of meta-cognition.

To know how serious the danger is we would have to have sample general principles of limitation-management and evaluate them against facts about human cognitive limits. But we don't have any serious such candidate principles, and we know very little that is useful about actual human cognitive limits {note 14}. But we can express a version of the danger in terms of computational complexity rather than human psychological limits, and then we can prove that the danger is real. I shall give two relevant results {proofs are in an appendix}.

For the first, assume a classification of algorithms into 'hard' and 'easy'. (Assume it applies to all algorithms. For example hard and easy might be non-recursive and recursive, or recursively enumerable and recursive, or nondeterministic polynomial time and polynomial time.) An algorithm will be hard or easy with respect to some domain of problems. (Any algorithm will be coextensive with an easy algorithm - a table of answers - with respect to a finite domain.) Assume that the hard and easy contrast satisfies two conditions. (i) the composition of easy algorithms is easy. (ii) every instance of a hard algorithm is also an instance of an easy problem. Now suppose that we have an enumerable domain D of problem-instances, and an algorithm A that gives answers to members of D. Then these two assumptions entail that it is not easy to trace the boundary between the hard and the easy members of D. The gloss I would put on this result is: there is no easy way of knowing the best way of solving a problem-instance. (There always is a short cut, but there is no short cut to finding short cuts.) So the signal for embarking on a cognitive routine that will handle an instance efficiently is not going to be given by direct reasoning, or any other easy method.

Instead of focusing on problem-instances we can focus on problems and the routines that give answers to them. Suppose that an agent is considering whether to reason in a way that involves answers to some problem. She has an algorithm that she thinks will give correct-enough results, but she does not know if it will give them in a reasonable time. She might wonder if there is some short cut she can call on to help; it would tell her not what the answers are but how long or difficult it will be to get them. It seems intuitively unlikely that many such short cuts are available. It seems likely that it is usually a hard question how hard a question is. And in fact we can prove a formal result to this effect.

We are considering algorithms from some denumerable set A = $\{a_1, a_2, \ldots\}$. Assume that there is an ordering of algorithms in terms of 'harder than'. The ordering might be just a two-stage 'easy'<'hard' . The question we are asking is whether there can be a function F that tells you how hard the computation of a(n) will be. And the answer is No, if we assume that the algorithms in A are decomposable with respect to some subset B of A. That is, a calculation of each a for each m is equivalent to trying successively a series of easier algorithms in B, until eventually one yields an answer. Given decomposability (and some other less serious assumptions) we can prove that no such F can uniformly provide useful bounds on the computation of individual arguments. In other words, when A is decomposable then no general procedure for predicting the computational cost of members of A for particular arguments can be easier than the class of algorithms into which members of A can be decomposed. The assumption that A is decomposable may seem quite strong. It makes the relation between A and B a generalization of the relation between deterministic and non-deterministic computation, and of that between recursive and recursively enumerable functions. (The result is thus an analog of the unsolvability of the halting problem for Turing machines theorem: there is no manageable procedure for saying whether for a particular input a machine gives a bounded or an unbounded response.) I suspect that the decomposability assumption is satisfied in a wide range of cases, but I leave this question to better mathematicians than I.

There are surely many variations on these results, stating general ways in which

the easy cannot substitute for the hard. (Particularly valuable would be results giving limits on one's ability to know how good an *approximation* to a correct solution is {note 16}.) General short-cut-excluding results will be unlike most results in the theory of computational complexity by not referring to particular complexity classes, but rather any classes satisfying certain constraints. They amount to taking facts that would hold of many different classes and finding the weakest assumptions from which they follow. It is important for philosophical purposes to have some such results, to assess the relevance of facts about the complexity classes of standard computational complexity theory to wider issues of cognitive difficulty. Results such as these are relatively independent of particular cognitive architectures and of particular reasons for emphasizing particular measures of difficulty. In particular, they are one of the kinds of consideration we need if we want to engage with questions about the appropriate normative framework for discussing human beings faced with problems at the limits of their capacities. In fact, once one accepts that the vocabulary of epistemic virtues is a natural one for developing the ideas that stem from work such as that of Harman and Cherniak, the next step must concern the division of labor between explicit self-knowledge about one's limitations and acquired virtues, and no-short-cut results such as these are essential to this step. (Results about where short cuts *may* be expected would of course also be extremely valuable.)

This paper is concerned with the states that any limited agents need when they aim both to acquire information rapidly and to amass it into coherent bodies of belief. Hard computation is associated with checking for consistency, and in general with thinking that considers whole bodies of beliefs rather than connections between given pairs of beliefs. ("Is this belief explanatorily coherent with what else I believe?" and "Is this belief evidence against anything else I believe?" are in the absence of miraculous short cuts going to be just as hard as "is this belief consistent with my other beliefs?") And whenever we have reasoning that deals with hard questions of this kind we will not be able to determine whether it is appropriate by answering easier questions. You can't count on short cuts.

Consider a very demanding epistemic ideal according to which an agent performs

all her belief-acquisition, all her checking for consistency, and all her immediate and longer-term repair of her bodies of belief in terms of direct and precise inference, whose appropriateness the agent herself can by means that the agent herself can determine, by applying general principles of rationality to her knowledge of her cognitive situation. Call this "crude internalism". But as should be clear by now crude internalism plus holism leads to cognitive overload. The alternative externalist mode is more helpful in such cases. A process is evaluated in terms of its propensity for success in situations in which the agent finds herself, whether or not she knows she is in those situations. Epistemic virtues, in particular, are appropriately deployed when their prospective components, given the situation as it is, lead the agent down cognitive paths in which their operational components result in true, useful, explanatory, or other aimed-at beliefs. The agent will in general not know what virtues she has, and whether she is deploying them effectively. According to this point of view we will always need an element of trust: sometimes we will simply have to follow the reasoning strategies that come to us, and trust our capacities for modifying them in the light of success and failure. That is, we will always need to have the prospective virtues that initiate lines of thinking that our operational virtues can then make good. And, the central point of this section, we can know in advance that we will usually not be able to know when a virtue will pay off {note 17} .

**5. No miracles** Limited rationality is not the same as irrationality or stupidity, of course, any more than limited moral insight is the same as evil. It just means 'like us'. According to the general picture that I have been developing, agents with limited rational powers need capacities, to which I think the term 'epistemic virtue' applies very naturally, which steer them through problems which are too hard to tackle with direct reasoning. Apart from some remarks on the inevitability of approximation and heuristics, I have said nothing about how we could instantiate such virtues. The picture is not a very helpful one if it requires us to manage our limitations by appeal to super-human powers. So how might we manage the cognitive routines in question?

I think the answer is very straightforward, and appeals to a traditional aspect of virtues, explicit from Aristotle onwards, that they are learned over a period of time from varied experience, good influences, and the emulation of role models {note 18}. The crucial fact is that the limits on a person's capacity to think through a problem in a given situation are a function of the resources, in particular time and memory, available in that situation. The things that are too hard for the person are the things that cannot be done within those time and memory constraints. But the situation is part of a longer term sequence, and over that longer term more complex things can be accomplished. The obvious greater resource here is time: an agent can accomplish tasks in preparation for a situation that she could not handle within it. In some cases we can think of this as a matter of proving lemmas or working out subroutines that then contribute to a stock that is ready to be applied to a variety of problems. In others we can think of it as thinking out conclusions about the efficiency, limits, and degree of approximation of cognitive routines, which cannot themselves be derived within the limits of a particular situation. Some of these conclusions, of course, concern the contexts and extents to which it will be best to rely on someone else's thinking or information.

One aspect of this is a familiar fact about computation. Often we can lessen demands on memory by making greater demands on time. A Turing machine with a short tape can compute some values of some functions that can be computed in a short time by a machine with a longer tape, as long as it is allowed to do a lot of shuttling back and forth along its tape. So if we regard the whole of a person's preparation for a task, in the limit the whole of a person's life, as a long computational process, we can see how in principle a person can without exceeding her cognitive resources at any point, prepare capacities which will help her negotiate problems that will have to be solved in a limited time window by using limited memory. As a description of actual human practice this seems to ignore important distinctions, in particular those between working memory, long-term factual memory, and learned dispositions. These if only we understood them better would add a realistic richness to the picture: the limits that constrain our capacity to think through particular problems in the contexts in which they

arise are largely those of working memory, and we manage these limits in part by preparing cognitive dispositions (virtues) and relevant facts (background belief) which can be applied to problems that we expect to arise. But in the present state of cognitive psychology there is very little that can usefully be said along these lines {note 19}. So these considerations must function in my argument simply as a possibility proof: there are ways in which the task can be accomplished. Nothing miraculous is needed.

A plausible speculation about the kind of learning that underlies epistemic virtues is that it has two components. In the first place we have a capacity to find patterns in particular kinds of problems. A person is exposed to some domain of tasks – basic algebra, making jokes, learning a language – and finds that they can develop ways of summoning the resources needed to deal with them. She begins to see what tasks are like what other tasks in terms of the applicability of a growing armory of prepared ways of thinking, ways of accessing memory, and ways of checking for errors. (Many people think that neural networks tend to be better at this kind of pattern recognition than proposition-manipulating routines.) Almost inevitably she does not develop this sense of how to apply her brain to many other domains, and the reasons that she is more successful with some than with others will be a mixture of deep facts about her intellectual resources and simple random accident. But given this beginning the second component can apply. It consists in seeing resemblances between problems in domains that are now familiar and manageable and new ones. From being good at algebra someone might go on to being good at making precise arguments; from being good at jokes someone might go on to being good at making up long serious stories; from being good at learning a language a person might go on to being good at understanding the history of some art form. Or they might not; none of these connections has any inevitability. The transfer of capacity from a familiar domain to a new one is far from automatic. The pattern recognition capacities get stretched and re-applied. Some people may find that for a variety of new domains they can fairly quickly find ways of re-applying their stock of cognitive tricks to them; others are best off sticking to what they're good at. We might think of the former class of people as

possessing epistemic meta-virtues: they have the valuable traits of being able to apply their virtues beyond their original domains {note 20}.

The important fact is that familiar aspects of scientific and everyday life are in principle explicable. People do master complicated intellectual domains, becoming able to work through problems in those domains with much less effort and time than equally intelligent other people who are good at different things. Typically we others can solve most of the problems that such specialized people can solve: but it will take us very much more time and effort, and we will rarely be sure that we have not made a mistake. And nearly all people find that sometimes when faced with totally new kinds of problems they can work their way into a feel for them that seems to work. And of course for many other new problems no such miracle occurs. And, as the model suggests, in nearly all of these situations it is quite opaque to the people concerned how they manage what they do. They can give very much less in the way of an articulate description of how they are proceeding than they can when explicit reasoning is the resource in question. Virtues are not typically introspectible.

These non-miraculousness arguments connect with issues about external and internal justification. A reason for following a certain procedure can be unavailable within a given context in that basing the procedure on it would require more resources than are available in that context. The procedure could all the same be in fact be a good one to use in the context, though the justification was external. In fact, the person might in principle be able to derive a justification, given more time and attention than is available in the context. I say 'in principle' because in almost all cases this would need more knowledge of human psychology and more understanding of the efficiency of cognitive processes than we have, perhaps than we will ever have. In practical terms, the point remains that a precise appreciation of how nearly optimal the procedure one is following is will nearly always be unavailable given any reasonable amount of reflection. But presumably one can sometimes from outside a context come to some rough sense that some ways of approaching some problems work well and others do not, which one can then call on to justify what one is doing. What counts as 'internal' and 'external' depends on where one draws the boundaries of the immediate

thinking subject {note 21}. Even this must be pretty rare in comparison with the times that what one is doing is in fact in accord with an effective strategy given the task and one's limitations, but one has no route to explicit knowledge of this fact. You often just have to trust yourself.

These conclusions suggest a shift in emphasis in the discussion of the internalism/externalism distinction in epistemology. Most discussions take the essential feature of states and processes susceptible to internalistic norms to be knowability: a person can know 'from the inside' whether they are in the relevant states. The significance of this information is that the person can then judge and correct her reasoning in the light of what she knows about her state and her grasp of the relevant principles of rationality. But the reflections just above and elsewhere in the paper suggest that the important fact is not whether one knows what state one is in but whether one knows whether it is in accord with some inferential principle that makes it likely to result in true belief. That fact, which on traditional accounts of rationality is easy to ascertain once one knows one's state, is in fact extremely hard to know. All the self-knowledge in the world will not help if you cannot tell where you should go from where you know you are. Then whether the idea is to start with internalistic norms and retreat to externalistic ones when it is impossible to apply them, or to start with externalistic ones and add internalistic ones as particularly useful special cases when they can be applied, the conclusion is the same. The significant frontier between the two is a matter of when internalistic norms can be applied, and this is very often not a matter of knowing your cognitive state but of relating it to any norm that your cognitive capacities allow you to grasp {note 22}.

Whatever the form of justification that can be given for a procedure that copes with a person's situation, the procedure will embody virtues and metavirtues of the kinds I have described. The virtues that I have been describing are virtues of the efficient management of one's cognitive limitations. I believe that these are a particularly significant class of virtues, central among the virtues of intelligent activity. And I believe that by pushing the line of argument in this paper further we may be able

to show that epistemic metavirtues are more interesting philosophically than epistemic virtues. It is with metavirtues that we have a chance of describing states of mind that agents can profitably aim at. Whether or not these beliefs are true, capacities for efficient management of one's cognitive limitations are at any rate epistemic virtues, which clearly exist, and whose inevitability follows from very general facts about the range of cognitive difficulty of the problems we encounter. {note 23: acknowledgements}

ADAM MORTON
UNIVERSITY OF OKLAHOMA
adammorton@ou.edu

appendix one - appendix two - notes - references - top of document

## NOTES

**1** This is a special case of a familiar theme, that we may not get good advice for creatures with limited cognitive capacities by scaling down theories intended for ideally rational agents. For two very different ways of developing that theme see Rubinstein (1988) and chapter 15 'Transcending humanity' of Nussbaum (1990). The most general form of the meta-resource catch is found in Lipman (1991). Lipman proves that under quite general conditions there is a point at which an agent can reach an equilibrium between thoughts, thoughts about thoughts, and so on. However there is no general way of telling where this point is!

**2** Emotions and virtues often share names, as with courage. I discuss the connection in Morton (2002).....

**3** These are all implicit in the literature on virtue epistemology. See Hookway (1999),

Sosa (1991), Zagzebski (1996).

**4** Cherniak (1986), Harman (1986), (1999) chapter 1.

**5** Note that this argument accepts that the best way is the best way. There is a sense in which it is (ideally) rational. Indeed there is a sense in which it is rational for human beings, if only they could find it and follow it. In not denying the rationality of the ideal I stay on the right side of the arguments in Millgram (1991). Millgram argues that there is no best point to draw the line between "hard but required" and "too hard to be required". I agree: that's one reason there is a variety of virtues.

**6** It follows that means cannot be found by mechanically setting them between the extremes any more than they can by setting them at the extremes. So moral and other virtues normally presuppose epistemic virtues of knowing where between the extremes the limits of appropriate action lie in the situation at hand.

**7** See Smith (1986), Jackson and Pargetter (1986), Zimmerman (1996), chapter 6.

**8** The nature of these complexity classes would be essential to a fuller discussion of their suitability as universal measures of cognitive difficulty. See appendix one.

**9** Note a curious fact. At the level of complexity that 'fits' propositional logic it is inconsistency that is harder to determine than consistency. Consistency is NP and inconsistency may be harder. At the level of complexity usually applied to quantificational logic it is consistency that is harder. Quantificational validity with respect to finite models is like propositional logic, but at the higher levels of complexity: consistency is r.e. and inconsistency is not. I am assuming that if NP turns out to be not distinct from P, as no one believes but no one has disproved, this is an exception to the general pattern among complexity classes.

**10** See appendix one.

**11** The fact referred to in footnote 9 suggests a policy of checking first for truth functional consistency, putting question marks by items that do not easily yield a yes answer, and then if resources permit checking further among the truth functionally consistent items for quantificational inconsistency. For a spirited defense of the rationality of ignoring some contradictions see Foley (1993), especially chapter 4 sections 5 to 7. Preparation for repair is presumably one reason why we have beliefs in the form of indicative conditionals. On learning that Shakespeare did not write *Hamlet* I am ready with the conviction that if Shakespeare is not the author someone else is.

**12** On mental models see Johnson-Laird (1983). It is important to keep an open mind about the relation between such models and both the models of model theory in logic and the models invoked in the structural conception of theories in the philosophy of science.

**13** The relativity of epistemic rationality to the fine-tuning induced by one's larger purposes is a familiar theme by now. A classic source is Levi (1967). A more recent version is found in Maher (1993.)

**14** The little we do know mostly concerns limitations of short term memory. See Baddeley (1986) One deep difference between the human and machine cases is that most machine memory is all-purpose while in human beings it tends to be purpose specific. See Gathercole and Baddeley (1993) chapter 4, for surprising consequences of the specificity of kinds of short term memory. Still, the general difference between time and memory (space) still applies. The familiar contrast between accuracy and reaction time (latency) at a task is one manifestation of it. Neil Immerman suggests to me that human cognitive difficulty might be measured with a complexity class that modeled very high limits on parallel processing very low limits on speed.

**15** For proofs see appendix 2

**16** Knowing how good an approximation to an exact solution a heuristic will give, and under what conditions, is likely to be often a hard problem. This is an important question because to many very hard problems we have approximate solutions that give good answers except on a 'difficult' set of inputs. The first of the two results of this section suggests, but only suggests, that distinguishing the difficult from the cooperative inputs is rarely trivial. Computer scientists are of the opinion that this will usually be the case, but as far as I know there are no general results.

**17** Different arguments for similar conclusions are common in the literature. I can cite only a selection of recent ones. For the irrelevance of internalist justification to real agents see Goldman (1999). For the way that virtues can fill a gap where no graspable rule can apply see Greco (2001). For the unknowability of the applicability of a rule see Williamson (2000) chapters 4 and 8.

**18** See Aristotle, Nichomachean Ethics Book II section 6, Zagzebski (1996) especially pp 102-106.

**19** We have some knowledge of how working memory increases in childhood and how it is related to linguistic development. See Gathercole and Baddeley (1993). What we know very little about is the ways in which limitations of working memory are circumvented by learned routines and facts stored in long-term memory.

**20** This picture emerged in conversations with Neil Immerman and Shlomo Silberstein about the AI analogs of these problems. There is a general similarity with the methodology of case-based reasoning. See Leake (1998).

**21** I am drawing here on a point in Sosa (1999), that there is no principled difference between chains of justification that fail to be contained in one person's cognition and

chains that fail to be contained in a short time span within one person. This point is also made in Goldman (1999).

**22** The inevitability of approximation makes the picture even more subtle. Suppose that a person is thinking in accordance with a routine that will often give adequate results but will in some circumstances lead her seriously astray. Suppose that she has no way of knowing how good her approximation is and how much danger she is in, as footnote 16 suggests is often inevitable. Is an internalistic norm appropriate. (Is she justified?) My philosophy-induced intuitions on the matter are contradictory. So much the worse for internalism, I would say.

**23** I have had a lot of help with this paper. Thanks to Luc Bovens, David Christensen, Ray Elugardo, Neil Immerman, James Hawthorne, Wilfrid Hodges, Danny Korman, Hilary Kornblith, Douglas Kutach, Chris Swoyer, Michael Williams, and Shlomo Zilberstein. I had really good discussions of drafts at York (BSPS), the University of Vermont, Paris (CREA), and the University of Colorado. The referees for Nous made observations which clarified a number of points for me.

appendix one - appendix two - notes - references - top of document

**REFERENCES**

Baddeley, A.D (1986) *Working Memory,* Oxford University Press.

Cherniak, Christopher (1986) *Minimal Rationality,* MIT Press.

Foley, Richard (1993) *Working without a net,* Oxford University Press.

Gathercole, Susan and Alan Baddeley (1993) *Working memory and language,* Lawrence Erlbaum Associates.

Goldman Alvin (1992) "Reliablism," in Jonathan Dancy and Ernest Sosa *A Companion to Epistemology,* Blackwell.

Goldman Alvin (1999) "Internalism exposed," *Journal of philosophy* **6**, 1999, pp 271-93.

Greco, John (2001) "Virtues and rules in epistemology," in Abrol Fairweather and Linda Zagzebski, eds. *Virtue epistemology: essays on epistemic virtue and responsibility,* Oxford University Press, pp 117-141.

Harman, Gilbert (1986) *Change in View,* MIT Press.

Harman, Gilbert (1999) *Reasoning, meaning, and mind,* Oxford University Press.

Hookway, Christopher (1999) "Epistemic norms and theoretical deliberation," *Ratio 12,* 1999, pp. *380-398.*

Immerman, Neil (1999) *Descriptive Complexity*, Springer.

Jackson, Frank and Robert Pargetter (1986) "Oughts, Options, and Actualism," *Philosophical Review* **95**, 1986, pp. 233-255.

Johnson-Laird, Philip (1983) *Mental Models,* Harvard University Press.

Levi, Isaac (1967) *Gambling with Truth*, Knopf.

Leake, David B (1998) "Case-based reasoning," in William Bechtel and George Graham. eds. *A companion to cognitive science,* Blackwell Publishers.

Lipman, Barton (1991) "How to decide how to decide how to …: Modeling Limited Rationality," *Econometrica,* **59,** pp. 1105-1125.

Maher, Patrick (1993) *Betting on theories,* Cambridge University Press.

Millgram, Elijah (1991) "Harman's hardness arguments," *Pacific Philosophical Quarterly* **72**, pp 181-202.

Morton, Adam (2002) "Beware stories: emotions and virtues," in Peter Goldie, ed. *Understanding Emotions* Ashgate, pp 55-63.

Nussbaum, Martha (1990) *Love's knowledge,* Oxford University Press

Papadimitriou, C. (1994) *Computational Complexity,* Addison-Wesley, 1994

Rubinstein, Ariel (1988) *Modelling bounded rationality*, MIT Press.

Smith, Holly (1986) "Moral realism, moral conflict and compound acts," *Journal of Philosophy*, **83**, 1986, pp 341- 360.

Sosa, Ernest (1991) *Knowledge in perspective*, Cambridge University Press.

Sosa, Ernest (1999) "Skepticism and the internal/external divide," in *The Blackwell guide to epistemology,* Blackwell Publishers, pp. 145-157.

Williamson, Timothy (2000) *Knowledge and its limits,* Oxford University Press

Zagzebski, Linda (1996) *Virtues of the mind,* Cambridge University Press.

Zimmerman, Michael J. (1996) *The structure of moral obligation,* Cambridge University Press.

appendix one - appendix two - notes - references - top of document

## Appendix one: the naturalness of complexity classes

The arguments in this paper draw on computer science, in particular on the theory of computational complexity. Computers work in very different ways to human brains, and the differences are likely to be particularly important when we consider questions of limitations: when we are looking for general and yet useful approaches to the ways in which performance is sensitive to varying amounts of certain basic resources. A superficial difference arises simply from our different interests in computers and in human thought. Computers deal quickly with tasks that take humans a long time, and so we think of them as instruments for which speed is of primary importance, and we measure the difficulty for a computer in terms of the time it takes to solve it. The more we rely on computers to do the things that take us too long to think out the more our emphasis in the human case focuses on more human specific aspects, particularly the contrast between tasks that a human can perform within a very short time using working memory and the tasks that we have to mull over using various kinds of medium and long term memory. The specific structure of human memory is likely to shape any discussion of human cognitive limitations that is relevant to the question "how we are to manage our intellectual ambitions given that some problems are to hard for us and many are too hard for us unless we tackle them in just the right way?" So are concepts drawn from computer science going to be any help at all?

I am sure that in the end any enlightening approach to issues about human cognitive limitations is going to differ in fundamental ways from the theory of computational complexity. (Just as the full theory of the limitations of digital computers

running on conventional silicon chips will have deep differences from the full theory of the limitations of analog computers running in biological media, even when we think of both in terms of carrying out algorithms as instruments in human society.) Yet, I am arguing, there are aspects of the theory of computational complexity that capture very general and universal features of how problem solving power can be more or less restricted, which we can use as part of a framework for articulating the limitations of human cognitive capacity. The claim here is thus in a general way analogous to Church's thesis, which asserts that a general model of computation can represent the limits of what any finite agent can calculate: so too, I propose, there is a general account of how the power of a finite but unlimited agent diminishes as the resources available to it are restricted, which can be meaningfully applied both to computers and to humans.

The core of the theory of computational complexity concerns the amount of time a particular algorithm as implemented on a particular machine takes to provide an answer to a question, and the amount of memory – 'space' - it requires. No distinction is made between kinds of memory. The main objects of interest are the problems, and the aim is to classify them into degrees of difficulty. A typical problem requires answers for all values of an integer variable n, so the concern is how the time or space required to come up with the answer varies as a function of n. A hard problem is one for which the function is a rapidly increasing function of the size of the input n, an easy one is one for which the function increases not much more steeply than the input. To get from this to something precise the sizes of the inputs and of the resources have to be characterized. There are no completely natural units for them, and the comparisons will be sensitive to choice of units. As a result computer scientists are interested not in exact measures of difficulty but in *complexity classes*. A complexity class is defined by a category of functions from inputs to resources. The most frequently used categories fall into a series: logarithmic, polynomial, exponential. (The next category would presumably be of the form k exp (s exp n), but there seems to be no practical need for this.) Problems in a complexity class are solvable by use of resources that are limited by functions in that class; for example problems solvable in polynomial time will get

answers such that for input n the answer will require f(n) units of time to compute, where f is a polynomial. Membership of a complexity class gives some independence from choice of units and measurement system (the greater the rate of increase of functions in the defining category, the greater the independence.)

(I fear my exposition will cut too many corners to please the experts, and yet have enough detail to baffle non-experts. For systematic and detailed expositions of the theory of computational complexity see Papadimitriou (1994), Immerman (1999). Papadimitriou is more self-contained, but Immerman makes the connections with logic that will interest philosophers.)

We thus have a base set of complexity classes, each consisting of problems that can be solved within either time or space bounds as set by a category of functions in the series. A particularly significant relation between classes is that between polynomial and exponential, and in general between a complexity class and that got by allowing exponentially more resources. For many algorithms require searching among the branches of a tree of possibilities, and searching among all the branches of a finitely branching tree will take at most exponentially many steps as searching along just one branch. Another, closely related, relation is that between deterministic and non-deterministic computation. For any class of 'deterministic' algorithms which proceed one step at a time one can think of a variant class of algorithms which at each step make a finite number of next steps, and then are regarded as having solved the problem if some series of steps led to the solution. This is the *non-deterministic* version of the original class of algorithms. Obviously any non-deterministic algorithm can be turned into a deterministic one that zigzags among the branches of the computations made by the indeterministic one, eventually covering all the same stages; but this will equally obviously take much longer, if no short-cuts are available. So for every complexity class there is another composed of problems which can be solved by nondeterministic algorithms within the same time or space constraints. In particular, corresponding to polynomial time complexity (P) there is non-deterministic polynomial time complexity (NP). All problems in NP can be solved in either polynomial or exponential time (there is nothing in between), but we do not know whether there are NP problems that require exponential time.

The contrast between deterministic and non-deterministic computation may not seem particularly significant. The contrast between computation within some constraint and computation needing exponentially more resources may seem more fundamental. But we generally do not know which problems require exponentially more resources than which others, while we do know which problems are non-deterministic versions of which others. In fact, the significance of the time and space (memory) complexity measures and the deterministic/nondeterministic contrast is reinforced by considerations on the one hand about the classification of natural and interesting problems and on the other about the deep links between complexity classes and the logical form of problems. The first of these considerations consists in the fact that problems that have independent interest from a computational point of view nearly always turn out to be maximal in one of the complexity classes one gets by attending to time, space, and the deterministic/nondeterministic contrast. That is, they are nearly always "complete" in some class: any other problem in that class can be reduced to them. (So they nearly always turn out to complete in logarithmic, polynomial or nondeterministic polynomial time or space complexity.) In particular the class of questions that are complete for nondeterministic polynomial time includes many interesting hard questions that elude efforts to supply manageable algorithms for them, such as the traveling salesman problem and the consistency (satisfiability) problem for propositional logic. Non-complete members of all these classes can be constructed, but they rarely have any interest apart from this fact. On the other hand, the relation between nondeterministic and exponential complexity is thoroughly mysterious. Any problem that can be solved in deterministic or nondeterministic polynomial time can clearly be solved in exponential time, but the question whether all problems that require nondeterministic polynomial time require exponential time, whether the inclusion $P \subset NP$ is strict, is the great lingering mystery of the subject.

(It is also a mystery, though not really a mathematical one, why the problems of independent interest tend to be complete in some otherwise interesting complexity class.)

The second consideration reinforcing the significance of the deterministic/nondeterministic contrast concerns relations with logic and formal languages. The fundamental observation here is that basic time and space complexity

classes can usually be linked to formal languages: for most such classes there is a formal language such that each problem in the class can be identified with the set of finite models for a sentence in that language. Thus first order languages correspond to what can be done in constant parallel time, first order languages augmented with the minimum apparatus for making inductive definitions correspond to polynomial time, and second order languages with only existential second order quantifiers correspond to non-deterministic polynomial time. These results are part of a general program 'structural complexity' whose aim is to connect complexity classes originally defined in terms of images of Turing machines operating under various limitations with classes of formal languages. To the extent that the program can succeed we can begin to glimpse something universal about these classes. It becomes much more plausible that these represent deep and universal aspects of cognitive difficulty.

(Proofs of these results can be found in Immerman (1999). The characterization of NP with existential second order languages is Fagin's theorem, which shaped the current direction of the subject. When I say that a problem can 'be identified with' a set of models I am alluding to a straightforward relation that takes quite a lot of detail to specify exactly.)

What are we to make this from the point of view of finite agents in general, human agents in particular? It does not seem likely that there is a most significant complexity class. The importance in computer science to polynomial time arises from the fact that what we want computers for is primarily to give us answers in a hurry, and many problems for which computers provide useful solutions, or for which we hope for good solutions, fall into classes for which the best available solutions lie within polynomial time. So there is a connection between polynomial time and the typical architectures of physical computers. But architecture is parochial: it does not follow that that polynomial time is a particularly significant complexity class for other kinds of computation, for example those performed by human brains. The equivalent of polynomial time for human cognition ought to be some class that recognizes the human characteristic of operating very slowly in a massively parallel fashion. And as for space, we would want the significant classes for human cognition to reflect the ways in which we stratify memory into different levels of permanence, as if there were two opposed expenses to be minimized: that of holding an item in immediate memory and that of

transferring it to longer-term storage.

Language on the other hand is less parochial; the logical forms that structural complexity correlates with complexity classes are ones that will be relevant to the thinking of an enormous range of conceivable thinking agents. An important distinction here, I think, is between processes and problems. The relation between deterministic and nondeterministic computation emerges from complexity theory as a relation between problems: some are harder than others in a way captured by the distinction. So the characterization of polynomial time in terms of first order languages plus inductive definitions does give it some claim to a universal significance. Universally harder problems ought to be harder for all finite thinkers. The tentative, very tentative, conclusion that I would suggest is that we can look at some of the simpler complexity classes as having some relevance for human cognition, and that some of the relations between classes of problems, particularly the deterministic/nondeterministic relation, are likely to correspond to factors which the structure of human cognition must accommodate.

The claim to universality of the deterministic/nondeterministic contrast is reinforced by its association with the recursive/recursively enumerable contrast. This association must be taken carefully though. (It is rather subtler than the main text of the paper may have suggested.) Any problem that can be solved by a nondeterministic Turing machine can also be solved by a deterministic Turing machine, which traces out a single path eventually covering the whole tree of computations of the nondeterministic machine. So recursive enumerability is not "nondeterministic recursiveness". A set is recursively enumerable, rather, when membership in it can be decided by a Turing machine augmented with the capacity to look ahead and tell when a branch of a computation will yield an answer within a finite number of steps. And this is a generalization of nondeterminism, since a problem can be solved nondeterministically when it can be decided by a deterministic machine with the capacity to look ahead and tell when a branch of a computation will yield an answer in 'not too many' steps. (What counts as too many depends on the complexity class in question.) But it is a generalization rather than an instance of it.

(Nondeterminism would be somewhat less significant, if it turned out that, against everyone's expectations, that NP is P. But this would still be a very special case, since the polynomial case is sandwiched between cases in an analogous inequality definitely holds. 'Above' we have the recursive/recursively enumerable case, and 'below', considering for example extreme restrictions on resources, the class of questions that can be answered deterministically in 17 steps is trivially a smaller class than that of questions that can be answered non-deterministically in 17 steps.)

It would be wrong to present very confident conclusions. The appearances are strong that deterministic and nondeterministic computation, branch-exploration and tree-exploration, deduction and consistency, are of different orders of difficulty. Few would doubt that the gap between deduction and consistency can be taken as something that will occur in computers, humans, and almost anything that thinks. But, more generally, we can understand the "harder-than" relation between problems in terms of increasing demands on time and memory, including human time and memory, in at least rough analogy to the tradition in computer science.

appendix one - appendix two - notes - references - top of document

## Appendix two: proofs

**(1)** Assume a classification of all algorithms into "hard" and "easy". (For example hard and easy might be non-recursive and recursive, or recursively enumerable and recursive, or non-polynomial time and polynomial time.) An algorithm will be hard or easy with respect to some domain of problems. (Any algorithm will be coextensive with an easy algorithm - a table of answers - with respect to a finite domain.) Assume that the hard and easy contrast satisfies two conditions. (i) the composition of easy algorithms is easy. (ii) every instance of a hard algorithm is also an instance of an easy problem. That is, for every algorithm A and every input i there is an algorithm A' such that A' is easy and A'(i) = A(i). Now suppose that we have an enumerable domain D of problem-instances, and an algorithm A that gives answers to members of D. Then these two assumptions entail that it is not easy to trace the boundary between the hard and the easy members of D. For suppose that there was an algorithm F such that for any dεD F(d) is a number representing an easiest algorithm for the d'th problem-

instance. Let S(d,a) be the result of applying the a'th algorithm to d. Assume that F is easy. By (ii) for any d F(d) is easy, and thus by (i) the composition of S and F is easy. But the composition of S and F is coextensive with A. So A is coextensive with an easy algorithm, which proceeds by taking an input d, finding the easiest algorithm which gives the same answer as A for d, and applying it to d. But A was an arbitrary algorithm, so if any members of D are hard F must be hard.

**(2)** Consider algorithms from some denumerable set A = $\{a_1,a_2, \ldots\}$. I shall take them to be indexed in such a way that we can conflate an algorithm $a_i$ and its index integer i. Assume that there is an ordering of algorithms in terms of 'harder than'. The ordering might be just a two-stage 'easy'<'hard' .

Suppose that there is an ordered set S and a function F such that F(i,n) = s, where s $\varepsilon$ S, iff the computation of $a_i(n)$ lies within bounds s. F thus tells you how hard the computation of $a_i(n)$ will be. (If the measure is time or space as normally conceived then s ranges over the integers, but we might want a partially ordered set for pairs of time and space, etc.)

Assume that one algorithm $a_i$ in A is harder than another $a_j$ iff there is a n such for all m>n F(i,m) > F(j,m). (Note that > is representing both the ordering of the integers and that of S.)

Assume that the algorithms in A are <u>decomposable</u> with respect to some subset B of A. That is, a calculation of each a for each m is equivalent to trying successively a series of algorithms in B, until eventually one yields an answer. More formally, assume three things. (i) for each $a_i$ in A there is a set $G_i = \{g_1, g_2,\ldots\} \subseteq$ B such that for each n there is a j such that $a_i(n) = g_j(n)$. (ii) for each $a_i$ there is a $g_j$ of maximum hardness, and there is a **h** in A such that all members of all $G_i$ are less hard than **h**. (iii) the composition of any algorithm in B with any algorithm no harder than an algorithm in B is also less hard than **h**. (Call (iii) the first composition assumption.)

Assume that the bound-giving function F is related to the decomposition of $a_i$ into $G_i$ in that the bound on the calculation of an $a_i(n)$ and the identification of the $g_j$ such that $a_i(n) = g_j(n)$ are easily recoverable from one another: there is a R such that F(i,n)

= s iff $a_i(n) = g_r(n)$ where r = R(i,s), and moreover the composition of R and functions in B is always bounded by **h** (call this the second composition assumption).

With all that in place, we can prove that no such F can uniformly provide useful bounds on the computation of individual arguments. For any $a_i$ consider the 'short cut' algorithm $s_i(n) = g_r(n)$ where r = R(i,s) = R(i, F(i,n)). For any n $s_i(n) = a_i(n)$. Now $s_i$ is a simple composition of F and R, which is then composed selectively with members of $G_i$: to get $s_i(n)$ you first compute F(i,n) then transform the result by R to get the the appropriate $g_r$ and then apply this $g_r$ to n. Assume that F is less hard than **h**. Then by the second composition assumption the composition of F and S is less hard than **h**. The composition of this with any $g_r$ will be less hard than its composition with the $g_j$ of maximum hardness, so by the first composition assumption will be less hard than **h**. So if F is less hard than **h** then there is an algorithm coextensive with $a_i$ that is less hard than **h**. But $a_i$ was an arbitrary member of A, so if F is less hard than **h** then all members of A are coextensive with algorithms that are less hard than **h**.

appendix one - appendix two - notes - references - top of document