How – and, What – Do Minds and Brains Optimize?

A Brainware Compatible Economics of Mental Behavior

2013 Andrew Brook Distinguished Lecture

Institute of Cognitive Sciences

Ottawa

Mihnea Moldoveanu

University of Toronto

April, 2013

Synopsis

I build a brainware-compatible 'modeling framework' for the economics of mental behavior – including perception, cognition in its many forms and the material pre-conditions for voluntary and involuntary action. 'Modeling framework' is in quotations to highlight a specific use of the word 'model' and its derivatives that will be developed herein, and which emphasizes purposive intervention and control as regulative goals, as opposed to description, explanation or representation. The resulting set of models make use of both the maximization- extremization protocols used by economists and 'neuro-economists' to derive behavioral predictions on the basis of agent-level 'utility', and the computational/symbolic representations of mental behavior used in artificial intelligence and cognitive science to represent mental states via symbolic structures, operations acting upon them and 'cognitive problems' and search processes. The new modeling framework is not an unfamiliar one. I use it to re-conceptualize 'what humans do' when they do what they do, to refine the set objective functions that are plausibly attributable to human agents and implementable on what we currently understand to be their brains.

1. "Human agents" (a phrase which stands for a variety of models and representations of real humans that share certain topological and logical features) "optimize" in the following sense: their behavior is represented as the outcome of one or more choices which under certain conditions can be represented as being the outcome of processes of optimization of some objective (utility) function. This constitutes an explanatory-predictive schema used in microeconomics to impute or ascribe objectives to (real) people on the basis of their observed choice behaviors, and to make predictions about subsequent choice behaviors on the basis of inferences made from their past choice behaviors to stable objective functions that in turn safeguard inferences to future choice behaviors.

1.1. For example, if Adam (a real person) chooses white bread when wheat bread is also available, then he is, in virtue of being represented as a rational agent - inferred to prefer (or, weakly prefer) white bread to wheat bread and therefore he is inferred to not choose wheat bread in the future in a situation ('from an option menu') in which white bread is also available.

1.1.1. That Adam's preferences do not change over time is a logically necessary condition for the inference from past to future behaviour, but is not 'tested against Adam's past behaviors', but, rather, posited as being constitutive of nature of a rational agent and therefore regulative of Adam's behaviour in virtue of him embodying such an agent.

1.2. There is a set of conditions on choice functions (namely: asymmetry, reflexivity, acyclicity/transitivity, completeness) which guarantee that if an agent's choice patterns satisfy these conditions in a domain of options, then there exists a real valued function that the agent can be represented as maximizing in making the choice she does.

1.2.1. For example, if agent A chooses x over option over option y, then (by revealed preference) one infers that she will not choose y over x is, and, moreover, that if his or her choices over x, y and other options $\{Z_i\}$ satisfy the set of conditions (asymmetry, acyclicity, reflexivity, antisymmetry) then there exists a real valued function $U(.)$ such that $U(x)>U(y)$.

1.2.2. To borrow a distinction from Kant: these conditions are constitutive of a rational agent, and regulative of the behaviour of a real person in virtue of that person being represented as a rational agent.

1.2.2.1. The passage from a constitutive condition to a regulative condition is (uneasily) safeguarded by a common commitment to rationality in its technical sense as a behavioural ideal. This commitment is not explicitly made by most of those who hold it, which accounts for the ambiguity and fuzziness that surrounds it in (largely very loose) discussions about 'whether or not people are/should be rational'. Although it is not my intent to air out this unclean linen of rational choice theory, what follows is not irrelevant to those who might want to do so.

1.3. This deductive schema, wherein the dynamics of a phenomenon is explained on the basis of the extremization of a scalar, real-valued multi-variable function *U(.)* whose existence is guaranteed by conditions that are constitutive of that system is common to the economic sciences in their axiomatic (choice-theoretic) form, but is also familiar to those who study the behavior of classical mechanical systems (potential energy), control systems (the Lyapounov function), large-scale stochastically evolving physical systems (free energy, entropy), biological systems (free energy, thermodynamic depth) and perceptual-cognitive systems (conditional entropy, model-conditional free energy, time-complexity).

1.3.1. Unlike the focus of the economic sciences on the axiomatic structure of the conditions under which the behavior of the object of study ('real people') can be said to instantiate the maximization of an objective function whose existence is constitutive of a model of that object (rational agents), the focus in other fields that make use of an extremization schemata for their explanatory apparatus focus on deriving 'equations of motion' that also model the *process* by which the (model of the) system in question carries out the extremization process.

1.3.1.1. Thus, 'friction-free gravitational free fall' in classical mechanics is characterized by the conversion of potential energy ($U=m(mass) \times g(gravitational\ constant) \times h\ (height\ of\ drop)$) into kinetic energy ($E=1/2 \times m\ v_f^2\ (final\ velocity)$) via fall along a geodesic (straight line in Euclidean space) (free energy minimization) extending from release to impact. More generally, 'equations of motion' for one or more classical particles can be derived from the Hamiltonian (potential energy) by taking the relevant first and second partial derivatives with respect to the phase space variables of the system, which yields a set of equations describing the phase space 'trajectories' of the system.

1.4. The deductive schema used in the microanalytic foundations of economics is moot regarding the process by which a 'rational agent' – a stylized model of either an ideal

or average human agent - carries out the extremization of the objective function whose existence is safeguarded by the conformity of choice patterns to a set of axioms. 'Optimization' is implicitly represented in such models as being costless and/or instantaneous; or as being antecedent to the place at which the model 'starts to work' and wholly irrelevant therefore to the domain of applicability of the model.

1.4.1. This is a representational move with significant and unfortunate consequences for the models that are predicated on it. Several objections to the validity of this move may be raised – which in turn lead to difficulties on which the rational choice schema itself is impaled- as follows:

1.4.1.1. A formal objection due to Leonard Savage: Suppose one has to choose between option *A* and option *B*, which are thought by the decision maker to have different payoffs. However, option *A* and option *B* are logically and materially equivalent, in the sense that choosing option *A*, in conjunction with several conditions that are 'self-evident', logically and materially entails choosing option *B*.

1.4.1.1.1. Question: should the axioms of rational choice be modified to enjoin a rational person to perform the set of deductive operations necessary to discover all of the logical consequences of what she already knows?

1.4.1.1.2. Answer: So modifying the said axioms will equate rationality with logical omniscience. That is a condition which is either too severe (normative sense) or unrealistic (descriptive sense) to impose on rational persons. For example, it would require that knowledge of the Peano axioms of the number system requires an agent to know whether or not the Goldbach conjecture (namely: 'Every even natural number can be expressed as the sum of two prime numbers') is true. At present, no real person who knows the Peano axioms is known to know whether the Goldbach conjecture is true or false. But under requirements of logical omniscience this entails that no person who knows the Peano axioms is rational, which is an undesirable result that is relevant to both the normative and the descriptive dimensions of the model.

1.4.1.2. An objection from common sense based on a suggestion of Bart Lipmann: Suppose agent *A* who does not have access to a computer or a calculator must choose between a lottery *L* that pays *$10,000,000* with

probability *0.1* and *0* with probability *0.9,* and a lottery *M* that pays *$10,000,000* if the 8^{th} digit in the decimal expansion of the real number representing the square root of *2* is *7*, and *$0* otherwise. Surely whether or not we deem A to be *rational* on the basis of her decision between these lotteries should depend on what we know (knowledge implies truth) about what *A* knows regarding the method(s) by which the square root of 2 may be calculated - for instance, Newton's iterative algorithm for calculating the roots of algebraic equations (in this case, the roots of $F(x)=x^2-2=0$). And, even if we know that she knows the algorithm in its formulaic form, we must also know that she possesses the means to implement the algorithm in an amount of time that is shorter than the deadline within which she must choose. The procedure by which optimization is carried out matters to reasonable judgments about the rationality of the optimizing agent, as does our knowledge of his knowledge of that procedure. It matters to both what we call rationality and to the process by which we test for it.

1.4.1.2.1. Comment: what should also matter to the degree to which we think A is rational on the basis of observing her choice between the two lotteries above is also what we know, and what we know she knows, about the architecture she has at her disposal for carrying out the requisite calculations. For example, if A has a brain lesion affecting her pre-frontal cortex that is sufficiently localized such that she (a) knows that there exists an algorithm for calculating the square root of 2 'by hand', (b) remembers each step of the algorithm in the sense of being able to write it down when asked to do so, but (c) cannot actually apply the algorithm to the specific problem at hand by carrying out the mental operations prescribed by the algorithm, and therefore (d) chooses the lottery that involves no computation in spite of the fact that she knows that she could significantly improve the expected value of her decision by carrying out the required computation, then we would have to treat her case very differently from the situation in which she can only 'imperfectly recall' the algorithm, or 'she does not know' the algorithm.

1.4.1.2.2. Comment: We do not need hypothetical or counterfactual brain lesions to make the point that the architecture on which A is supposed to implement a computation that is logically and materially required for a 'rational decision' (whatever that might be) – and A's insight into this architecture – should matter to the degree of

rationality that we ascribe to A (or, to the structure of the model of rational choice we believe represents A). A need only have imperfect knowledge about the degree to which she can carry out algebraic computations 'without error', for it to be the case that, again, she may 'rationally' (the term is increasingly volatile) choose the lottery requiring no computation over that which does, in spite of the fact that she is well aware of Newton's method for computing the square root of 2.

1.5. The deductive schema by which individual behavior is explained as the outcome of choice that instantiates the outcome of an optimization process should be modified to include insights about the process by which optimization is carried out and the architecture – cortical and sub-cortical structure of human brains – on which it is carried out.

1.5.1. Why 'about the architecture on which optimization is carried out' as well?

1.5.1.1. Optimization is a procedure.

1.5.1.2. Procedures depend for their instantiation on a particular implementation or a physical realization.

1.5.1.3. Procedures 'run' on material substrates', in the sense that there is a correspondence between the advance of the procedure towards the optimal point and a set of physical events occurring in a bounded region of space-time.

1.5.1.4. The said region must fulfill a set of material preconditions required for a procedure to run on it. These pre-conditions are constitutive of that procedure in the same way as the conditions for rationality are constitutive of models of rational choice.

1.5.1.5. The parameters of the process by which a procedure runs are determined by the properties of the space-time region on which they run.

1.5.1.6. Models of such material conditions are constitutive of models of the ways in which optimization procedures are carried out.

1.5.1.7. Brains are the space-time regions on which humans' optimization procedures 'run'.

1.5.1.8. Models of brains are constitutive of models of the processes by which agents representing humans who possess brains optimize.

1.5.1.9. (Known, discovered, tested, introspected) properties of brains are constitutive of a model of optimization as applied to humans.

1.5.2. Arguments both pro and contra 'neuroeconomics' miss the point of the difficulty that physical realizability, and procedural definiteness and specificity raise for optimization-based models of human behavior.

1.5.2.1. Camerer, Prelec and Loewenstein [2008] are wrong in their insistence on including neurophysiological variables into economic models of choice behavior for the reason that this would increase the predictive accuracy of these models. Predictive accuracy alone cannot motivate a choice of specific predictor variables. Positing variables that encode environmental conditions that (may) influence agent level choice patterns *via* neurophysiologically intelligible mechanisms can serve the purpose of enhancing predictive accuracy of choice models just as well, as Gul and Pesendorfer [2005] correctly point out.

1.5.2.2. Gul and Pesendorfer [2005] are wrong in the specific assumption that the concomitants of choice (environmental conditions, option menus) behavior exhaust the range of variables that should matter to economic analysis and therefore that the procedures by which agents optimize and the architectures on which this optimization takes place – and their insights into the architecture – need not be included on pain of rational choice models being radically incomplete. The point of the examples above is that the very nature and form of the optimization process that is ascribed to an agent depend on the optimization processes and architectures that are constitutive of the said agent.

1.5.2.3. Bernheim [2007] is wrong in the specific assumption that out of sample predictions of choice behavior either require or would be facilitated by 'neuroeconomic' models of choice behavior, which constitute a valid reason for introducing brain-specific considerations into economic research. It is the internal coherence of optimization – based

explanations that is in jeopardy if procedural and architectural considerations are not introduced in microanalytic optimization-based models and it is concern for rectifying logical problems arising from this omission that should be the overriding concern on the basis of which such considerations should be introduced.

1.5.2.4. Bernheim [2007] is right in arguing for a logic of inquiry into the neural bases of decisions and choices that maps choice onto the outcome of decisions that are the outcomes of algorithmically structured processes that can be implemented on neurologically verified structures. Decisions instantiate maximal points – or optimal solutions – of decision problems, which are solved using algorithms which 'run' on cortical and sub-cortical structures. The reason why this is right, however, is that there is a logical problem with the adduction of optimization-based models of choices made by humans that this explanatory schema solves.

1.6. Mental behavior is behavior. As such it, too, is the outcome of choices and 'pulled by the net expected value of micro-local incentives' rather than 'pushed by causes'.

1.6.1. Mental behaviour refers to any identifiable temporal sequence of mental events, which include instances of perception, sensation, reasoning, remembering, acting.

1.6.1.1. They include the subjective experience we call 'pain'.

1.6.1.1.1. How is *that* pulled by incentives, when it (plainly) seems pushed by a cause (the noxious stimulus)?

1.6.1.1.2. While the interaction between the physical stimulus and the sensorium is 'pushed by causes', the mental behaviour called 'attending to pain' is not.

1.6.1.1.2.1. Real humans in real labs can choose whether or not to 'attend to pain', and can voluntarily turn off the cortical projections of the pain experience [deCharms, 2005].

1.6.1.1.2.2. They can therefore choose to cause themselves to not feel the pain that one who would not have so chosen would feel in an identical situation.

1.6.2. If the procedures that carry out optimization tasks and architectures on which these procedures run are relevant to optimization-based models of choice behavior, then these models need to spell out not only the specific functional forms of the optimization problems human agents carry out, the (algorithmic) forms of the procedures that solve these problems and the architectures on which these procedures run and converge in finite amounts of time, but also the objective functions that are extremized by agents when figuring out whether or not to solve a problem, which procedure to use when solving that problem, whether or not to think further about the problem via the implementation of a procedure by engaging - at the margin - in one additional mental operation that maps one mental state onto another, and, when to stop thinking altogether.

1.6.3. What is needed is *a local economics of mental behavior* that is informed by the cortical and sub-cortical structures on which this behavior supervenes, and which minimally specifies the marginal costs and benefits of mental behavior and the option set that an agent capable of producing mental behavior has at her disposal.

1.6.4. This insight, contributed by economics to neuroscience and cognitive science, radically transforms what we men by 'a model of thinking' or 'a model of cognition', or 'a model of emotion'.

1.6.4.1. Local (neurophysiological) incentives, algorithmic form and architectural conditions are all constitutive of such a model.

1.7. Seen through the prism of incentives and micro-local maximisands, mental behavior can be understood as a form of 'directed cognition', and 'thinking' – at least of a certain kind – has been so described [Gabaix and Laibson, 2005].

1.7.1. 'Directed' in the phrase 'directed cognition' begs the question as to what directs it.

1.7.2. Applying the microeconomic calculus of representing behavior as the outcome of choice patterns that follow certain conditions in virtue of maximizing some objective function – and vice-versa – at the *pico* level (thinking, cognition, etc) takes us to positing a value function which determines whether or not someone will continue to think 'another step'. If b_{t+T} represents the thinker's estimate of the marginal value at t of thinking for another time epoch of duration T (a real number: as if biological time is infinitely sub-divisible – my aside) and c

represents the marginal cost of thinking, then the thinker will think T time units further iff $\max_{R \ni T} E_t[V(b_{t+T}, t+T) - V(b_t, t)] - cT > 0.$ Increments in the instantaneous 'value of thinking further' come from the decrease in some reduction in the variance of the thinker's (subjective or objective) estimate of some quantity of interest.

1.7.2.1. For instance, the value of interest may be the exact value of the *1000* point decimal expansion of the square root of *101*, or the variable that encodes the expected value of the *k*th branch of an *m* stage decision tree.

1.7.2.2. In these cases – which are typical of those given in Gabaix and Laibson – what the thinker is supposed to 'do' within the quantum of time T within which the expected value of further thinking is evaluated is clear enough: it is determined (guided, prescribed, inscribed: let us not get hung up on this word) by a procedure or an algorithm that specifies an operation that mental energy shall be put into implementing.

1.7.2.2.1. But, the 'pico-economic' model of directed cognition does not specify the dependence of the expected value of incremental thinking on the specific form of the procedure that prescribes the temporal chaining of the 'think further' operator. 'Think forward how?'

1.7.2.3. Moreover, the directed cognition model also does not specify the specific architecture on which cognition is supposed to be implemented. Any architectural constraints 'come into the picture' through the specification of c, the marginal cost of thinking.

1.7.2.3.1. But, c will vary not only according to the 'hardware' – read 'brainware' – involved, but also in accordance with the types of operations that are prescribed by a putative cognitive procedure or algorithm.

1.7.2.3.2. The model has 'nowhere to go' in terms of specifying the dependence of the marginal cost and expected incremental benefit of thinking for T seconds more on the procedure by which thinking 'happens' and the architectures on which thinking happens.

1.8. 'Subjective or objective' presents a difficulty for the model. Both prongs of the disjunction are problematic.

1.8.1. If 'subjective' variance is minimized, then, how to differentiate between the thinker's estimate of how surprised he will be conditional upon thinking *T* units of time further that is based on an understanding of the computational dynamics of the problem, and an estimate which is equally subjective but arises from her local 'along the way anxiety' about not having gotten as much closer to a solution as she would have hoped?

1.8.2. There is also the pragmatic and logical problem that the thinker cannot know how surprised she will be by the next bit of mental activity she has engaged in, because she has not yet engaged in it, and there is no algorithm or procedure which is specified in the model, and which

1.8.3. If 'objective' variance' is minimized, then where, in the model, does the measurement, ascertainment or even computation of this variance fit? Notice that since the model models 'mental behavior ('directed cognition'), it should provide 'variable space' for the incorporation of the variables that the model itself makes reference to as relevant to the thinker. But, it does not.

1.8.4. The situation could be remedied by providing for a set of procedures or algorithms $\{A_k\}$ – with known performance estimates such as per-operation benefit (reduction in variance or spread of the thinker's probability distribution function for possible values of the answer) and per-operation cost (the thinker's 'physiological' disutility of engaging in an operation and the expected value loss of making a mistake). But then:

1.8.4.1. Algorithms can be given tight definitions for problems that are solvable on Turing Machines (TM) or Random Access Machines (RAM). Brains are not self-evidently adequately representable as either, without further work. This patch is therefore not self-evidently effective.

1.8.4.2. It is unlikely – even if not logically or materially impossible - that a (real) thinker will know the marginal and average performance characteristics (costs and benefits) of the problem solving procedures she is using. The patch therefore leads to a model that is *a priori* unlikely.

1.9. The problems and difficulties I have strewn out here are logical and conceptual on one hand and pragmatic on another, not "empirical", "methodological", or "epistemological".

1.9.1. They have empirical consequences, but their value should not be gauged by the success with which heeding them meets.

1.9.1.1. It is not merely empirical success that has made rational choice theory so appealing an instrument of representation and intervention.

1.9.1.1.1. It is certainly not empirical success in the form of 'better explanation' - where 'better' can be understood as 'goodness-of-fit-weighted generalizability of explanations of observation statements', or 'generalizability-weighted goodness-of-fit of explanations of observation statements'.

1.9.1.1.2. And to the extent that the value placed on 'better explanation' is based on the premise that better explanation leads to better prediction, the fact that this assumption is incorrect entails that the value of better explanation should be nil.

1.9.1.2. What makes rational choice theory additionally and perhaps uniquely valuable is its usefulness as a tool of intervention – of policy-making and mechanism design.

1.9.1.2.1. Representing real people as rational agents allows an auction designer to engineer mechanisms which enjoin self-interested, logically proficient individuals from inefficient but personally gainful appropriations of value.

1.9.1.2.1.1. The rational agents whose behavior the mechanism is designed to shape or constrain are sufficiently life-like for the mechanism designed on the premise of such models to effectively constrain the behaviors of real people. That they are indeed sufficiently life-like is an inference to the best explanation for why the mechanisms 'work as predicted', not an axiom of representation, nor an inductively corroborated 'law of human behavior'.

1.9.1.2.2. Representing real people as rational agents allows macro-economic policy makes to make predictions of the effects their actions will have on real-people, and thereby to design effective control and intervention strategies for interacting with real people.

1.9.1.2.3. The model allows the policy maker to intervene effectively, and an inference to the best explanation of why the intervention is effective is that the model is representationally successful.

1.9.1.2.3.1. "Representationally successful" means more than 'generative of accurate predictions'. It means that the terms of the model have genuine referents.

1.9.1.2.3.1.1. Thus Ian Hacking: "The reason why I think that electrons are real is because you can spray them."

1.9.1.2.3.1.1.1. Aside: Before becoming too enchanted with these words, we need to pay close attention to whom 'you' in the above statement is intended to refer to.

1.9.1.2.4. The pragmatic effectiveness of the model is relevant (at least) to the value of the representation it embodies.

1.9.1.2.5. We can increase the value of this representation by increasing the pragmatic effectiveness of the model.

1.9.1.2.6. We can do so by adding 'levers' to the model: i.e. 'things you can spray'.

1.9.1.2.7. Neuronal circuits are examples of 'things you can spray': you can excite them via trans-cranial magnetic fields, and ablate them using high power electrodes and laser diodes, for instance.

1.9.1.2.8. Neuronal circuits make up the architectures on which optimization 'runs'.

1.9.1.2.9. We can increase the value of optimization based models by adding variables and relationships reflecting the constraints imposed by the architecture on which optimization runs.

1.9.1.2.10. The fact that we can do so, coupled with the recognition of the do value of doing so, suggests that we should consider doing so. It does not entail we should do so, of course. That would only be the case if we additionally knew there is no better way of accomplishing what we want.

1.10. The foregoing discussion is meant to be illustrative and not dispositive of the difficulties we encounter when trying to build a useful economics of mental behavior.

2. Mental behavior – including cognition, perception and the processes underlying irreversible commitments to voluntary and involuntary action – are not adequately representable solely as context-free, rule-based, objective-free symbol manipulation procedures.

2.1. The study of problem solving in cognitive psychology and artificial intelligence proceeds as follows: given a knowledge base *K comprising a* problem *P* which encodes a mismatch between current and desired conditions for an agent X, with solution search space *S* containing solution *s*, and a rule set *R* that describes allowable transitions in states of *K*, a problem solving process is represented by a search in *S* that begins from an initial state *k* which is a proper subset of *K* (containing *S, R)* and proceeds, via a sequence of transformations of *k* according to subsets *r* of rules in *R*, to find *s*, provided that *s* is in *S*.

2.1.1. *K* thus comprises a set of symbols that the mental process *M* modeled by the *n-tuple* (*k: P,R,S)* manipulates via sequential application of *R*, where the symbols in question may be textual, linguistic or abstract entities that stand in a 'reference' relationship to sensory perceptions or assemblies thereof, or images or other non-linguistic signals that stand in a 'coding' relationship to sensory perceptions or assemblies thereof;

2.1.2. In a tighter formulation of the 'problem solving as structured symbolic manipulation' representation due to Herbert Simon [Simon, 1973], *K* comprises a description of the current and solution or goal states, a test to determine whether or not the solution state has been reached, a set of operators that act upon elements of *K* in ways that are bound by rules *R*, a set of descriptors for the intermediate states caused by sequential applications of these operators, a set of differences among relevant states of *K* and the solution state, and a connection map which links the said differences to one or more operators whose application to the current or intermediate set of states is likely, plausible, or known to reduce these differences.

2.1.2.1. A well-structured problem is one for which the solution search space *S* and the location of the solution *s* in *S* does not change in a way that is causally linked to the application of any operator to any state of *K*. An ill-structured problem is a problem that is not well-structured.

2.1.2.1.1. This might seem like a useful distinction, but it is often advanced in order to rule out ill-structured problems from the domain of

problems in the domain of inquiry of 'problem solving'. This has the unfortunate consequence that very interesting problems successfully solved by human agents – like balancing trays loaded with containers full of hot liquids – are not taken as seriously as they should be.

2.1.3. The structured-ness of a problem according to the definition(s) in 2.12.1 depends not only on the causal links between search and solution space topology and size of the solution space, but also on the degree to which the problem solver behaving according to the protocol of 2.1.2 'sees' the entire solution space before beginning to search for *s* in *S*. Absent logical omniscience and in the presence of combinatorial explosions of the cardinality of the solution space *S*, bounded (logically non-omniscient) search can lead to reasonable modification of *S* that result from applying some operator to the current state of *K*. Ill-structured-ness of a problem is therefore also contingent on the logical prowess of the problem solver.

2.1.3.1. 'Sees' should not be interpreted literally (not many words should). It is meant to stand in for a state at which the problem solver either (a) has enumerated, or (b) is in the possession of a fast procedure for enumerating, or (c) is in possession of a fast procedure for building a fast procedure for enumerating the elements of the solution space; or, (d) is in possession of a measure of the cardinality of the solution space, or (e) is in possession of a fast way of accurately estimating in a reliable fashion reliably the cardinality of the solution space of the problem. 'Sees' should be parsed as in the colloquial 'see how hard this is?'

2.1.3.2. This makes ill-structuredness even more interesting, since logical omniscience is not in a nay case a good modeling assumption when it comes to representing real problem solvers (humans or machines).

2.1.3.3. To the extent that well-structuredness depends on the degree of insight of the problem solver on the (computational) complexity of the problem she is about to solve, it does not seem plausible that most interesting problems are well-structured

2.1.4. The 'tighter' representation 2.1.2 does not directly address the sequencing of operators applied to *K* with the aim of moving from the current and intermediate states of the search process towards the solution *s*.

2.1.4.1. To be sure, one cannot 'minimize the difference' between the current state and a desired solution state without knowing what the solution state is, just like one cannot 'minimize surprise' before experiencing the event that would have produced it or 'minimize the variance of a search process' without sufficient statistics on the outcomes of similar search processes conditional on similar states.

2.1.4.2. Thus the optimisand that would presumably 'direct' the search process is uncomputable by the problem solving agent.

2.1.4.2.1. Even if this optimisand were computable, there is – again – 'no room' in the 'tighter' model of problem solving 2.1.2 for incorporating this computation.

2.1.4.2.1.1. The difficulty is similar to that encountered by someone who tries to press fit problems like 'Find – by the fastest process for doing so – the shortest route of getting from point A to point B' (P1) onto problems like 'find the shortest route that takes you from point A to point B' (P2). A maximally parsimonious model for P2 problem solving processes will not do well handling P1 problem solving processes.

2.1.5. A directed version of a computational model for problem solving, that takes into account the difficulty of providing local guidance during the search process which mimics the intentionality of intermediate stages of problem solving (in humans) is needed.

2.2. In a directed version of this representation, the sequential application of rules r in R to the structure k are guided by a problem solving procedure or algorithm $A{:}K \rightarrow K$, which prescribes the applications of rules r to K as a function of the instantaneous state k_m of K during the search process.

2.2.1. Decision Problems. Let Σ be an alphabet (eg. {a,b,c,…}), Σ^* be the set of all words that use all and only the symbols of Σ and $L \subseteq \Sigma^*$ be a language that exclusively uses the set of words Σ^*. Then a Decision Problem P_d is a triple L, Σ, Σ^* and algorithm A solves the problem P if for every input $x \in \Sigma^* A(x) = 1$ *if* $x \in L$ and $A(x)=0$ if $x \in \Sigma^* - L$. The algorithm A computes a function from Σ^*- the language in which x is expressed - to {0,1} – the output of the decision problem (0='no'; '1'='yes').

2.2.1.1. 'Rational' agents can be represented as solving decision problems in virtue of the fact that they 'make decisions'. Using this manner of speaking to represent what decision agents do when they make decisions, however, is unlikely to be satisfying to a modeler that wants to say that rational agents' choices reflect the optimization of some (objective, utility) function. To make the procedural component of the optimization process explicit, we need to introduce an 'optimization problem' in algorithmic form:

2.2.2. Optimization problems. An optimization problem P_O is *a 7-tuple* $(\sum_I, \sum_O, L, L_I, M, cost, goal)$, where

$\sum_I$ is an alphabet, called the input alphabet of P_O, $\sum_O$ is an alphabet, called the output alphabet *of* P_O, $L \subseteq \sum_I^*$ is the language of feasible problem instances, $L_I \subseteq L$ *is the* language of the (actual) problem instances of P_O, M is the function from L to one of the $2^{|\Sigma o|}$ elements of the power set of $\sum_O$ and, for every $x \in L, M(x)$ is called the set of feasible solutions for x, *cost* is the cost function that, for every pair *(u, x),* where $u \in M(x)$for some $x \in L$, assigns a positive real number $cost(u, x), goal \in \{minimum, maximum\}$.For every $x \in L_I$, a feasible solution $y \in M(x)$ *is* called optimal for x and P_O if:

$$cost(y, x) = goal\,\{cost(z, x) | z \in M(x)\}.$$

For an optimal solution $y \in M(x)$, denote $cost(y, z)$by $Opt_P(x)$. P_O is called a maximization problem if *goal = maximum,* and a minimization problem *if goal = minimum.* $Output_P(x) \subseteq M(x)$ denotes the set of all optimal solutions for the instance x of P_O.

2.2.2.1. An algorithm A is consistent for P_O if, for every $x \in L_I$, the output $A(x) \in M(x)$.

2.2.2.2. An algorithm B solves P_O *if* B is consistent for P_O, and for every $x \in L_I, B(x)$ is an optimal solution for x and P_O.

2.2.3. An agent can 'have' (behold, attempt to solve) problems but not solution algorithms for them, and 2.2.1 and 2.2.2 allow for that.

2.2.3.1.1. Knowledge of a problem does not entail either knowledge of its solution algorithm or the existence thereof.

2.2.3.1.2. Knowledge of a solution algorithm entails knowledge of the problem it is designed to solve.

2.2.3.2. An agent cannot have (or, 'run') a solution algorithm without a problem it is solving, and 2.2.1 and 2.2.2 provide this restriction.

2.2.3.2.1. Representing agents as 'routines' – or, algorithms – that run independently of the problems the agents are solving makes no sense. It is akin to representing the behavior of a deterministic causal system as being generated by sequential choices made by components of the system along the way.

2.2.4. A problem – decision or optimization – has an infinite number of instantiations.

2.2.4.1. Example: The CLIQUE problem (Does Graph G(V,E), comprising vertices {V} and edges {E} have a clique (a fully connected sub-graph) of size *l*? has an infinite number of realizations, corresponding to all possible graphs.

2.2.4.2. Example: The Reachability Problem (Is node *i* in a digraph G(V,E) (a graph G comprising vertices {*V*} and edges {*E*} such that each vertex is directional) reachable from node *j* that is also in *G*, i.e. can one get from *i* to *j* using by traveling only along the edges of *G* only in the direction consistent with the directionality of these edges ?) has an infinite number of realizations, corresponding to all possible digraphs *G*.

2.2.5. An algorithm *A* for solving problem *P* has an infinite number of instantiations, corresponding to the instantiations of *P*, and the multiplicity of algorithms for solving each instantiation.

2.2.6. Problems and algorithms have intuitive interpretations in everyday examples that involve not only what some are call 'thinking', but also 'perception' and 'action'.

2.2.6.1. *P:* "Solve $ax+b=c$ " for *x"* is a prototypical example of what we are used to calling a 'problem'. The problem has a unique optimum (a closed form expression for *x* as a function of *a,b* and *c*), languages for inputs and outputs (variables ranging over the real number system, arithmetic operators) and a cost function whose goal is maximization (it takes on the value *1*) for the correct solution for *x*, and *0* for all other solutions). *A* consists of a sequence of actions whose application is guided by a set *r* of rules:

2.2.6.1.1. Step 1: Group all terms of identical order in the powers of *x* on the same side: $ax=c-b$

2.2.6.1.2. Step 2: Solve for x by dividing through by a: $x=(c-b)/a$.

2.2.6.1.3. If necessary, a verification step proceeds by substituting the solution for x ($x=(c-b)/a$)) in the initial equation $ax+b=c$ to check that indeed this value of x satisfies the equation, i.e. $a[(c-b)/a]+b=c$.

2.2.6.1.3.1. If necessary, each sub-step of the verification step can be prescribed by an algorithm whose operation-wise application depends only on the validity of the Peano axioms for the number system and the definition of addition, multiplication and their inverses (subtraction and division) on which the axioms are predicated.

2.2.6.2. *P="find the Nash equilibrium in a 2x2, 1*-shot, Prisoner's Dilemma Game' is another prototypical example of an optimization problem. There is an input language (players, strategies, payoffs, beliefs, conjectures; the axioms of set theory and the real number system), an output language of functions (*equilibria* – or, fixed points of the 'strategic mutual best response' mapping) that range over the primitives of the input language, a cost function (Pareto optimality, maximum) and a space of feasible solutions (the set of all strategy pairs and payoffs).

2.2.6.2.1. *A* consists of a sequence of (mental) actions (rank cells according to magnitude of outcome for row player, choose maximal outcome for column player that maximizes value of outcome of row player).

2.2.6.3. *P*='Find your (battery-drained-) Smart Phone in a (now-empty) hotel conference room' is an optimization problem that can be interpreted to involve both thinking and action – which in turn can include operations performed by muscle spindles and internal operations ('heeding', 'attending to'). There is an input language (roughly: the set of points comprising the space of the conference hall in a reference Euclidean coordinate system, the specific function relating the points corresponding to the location of the device in the space of the hall, a function describing the time-dependent motion of the owner of the device as she criss-crosses the hall in search of the device), an output language (the co-location of the owner and the device and the payoff) a solution search

space (all feasible – i.e. compatible with kinematic and structural constraints and the laws of physics) locations of the device within the hall).

2.2.6.3.1. A (possible) algorithm A consists of a series of basic search operations performed by the owner of the device that are meant to 'find the device' – i.e. to achieve co-location of the user and device within an area of at most *a*. These operations include – but are not limited to – patterns of walking around the conference center, patterns of scanning the local neighborhood in which the owner finds herself after *1,2,3..,n* walking steps, patterns of raising or lowering the field of vision ('squatting') in order to look under tables and chairs, and so forth.

2.2.6.3.1.1. A may be simulated (or, emulated, i.e. imagined, represented, visualized) by the problem solver, or it may be embodied (i.e. she is 'doing it').

2.2.6.3.1.1.1. This distinction requires careful thinking about what we mean by both a 'problem solving agent' and a 'problem solving procedure' – or, algorithm. Does an 'algorithm' refer to the series of instructions that comprise it or to the process by which a (suitably compiled) program embodying it runs on a piece of hardware (like, a brain)?

2.2.6.3.1.1.2. A computer program 'solves' for the eigenvalues of a large square matrix both in the sense that it is the program which, if run on an adequate device, can solve for its eigenvalues, and also in the sense that, when running, it actually produces the said eigenvalues as the output.

2.2.6.3.1.1.3. 'Solving for the Nash Equilibrium of a game' is clearly different from 'memorizing the sequence of instructions that correspond to solving for it'.

2.2.6.3.1.1.4. One can memorize the sequence of instructions – and even 'explain' what each instruction means in plain

old English, without being able to actually perform one or more of the instructions.

2.2.6.3.1.1.5. Performance need not entail error free performance. That is, one may be able to perform the specified instructions, but may only do so imperfectly. Whereas, the computation of the equilibrium of *this game* requires error-free – or, at least, error-proof, upon verification – implementation of the instructions.

2.2.6.4. The problem *P:* 'Balance this tray of containers containing hot liquids above your head with one hand' is an optimization problem that brings the contrast between embodied and 'cognized' algorithms into even sharper focus. The optimum is in this case not a real scalar or vector, but a function that maps different forces that the person doing the balancing 'act' senses – which measure and represent the instantaneous tilt of the tray – and a sequence of opposing forces – exerted through the palm, the wrist and the five digits supporting the tray – which restore the tray to (within some small quantity *e* of) its horizontal position. (As an exercise: construct the input and output languages, the search space). Clearly, the search space is very large, if it is bounded at all (the point forces are real numbers, and the number of ways (combinations, permutations, at different levels for each distinct point of contact) in which forces may be applied through the points of contact between the hand and the tray is very large.

2.2.6.4.1. 'Optimization' in this case is clearly not something that one does 'offline' – in purely cognitive terms – and then applies 'online' – in the way of a physical embodiment. It is 'embodied all the way'.

2.2.6.4.2. The problem also highlights the problem of matching the time constant of 'computing a response' to the tilting tray and that of the tray falling (along the path of least resistance). A 'balancing algorithm' will only work if it world 'quickly enough' – i.e. if it produces an action that counteracts the relevant forces in the right amount of time, otherwise the tray will fall. In this case, it makes sense to think of optimization as both local and bounded – by the time available to perform it, and by the computational resources that can be deployed within the maximum allowable time.

2.2.6.5. The problem *P: Produce a facial expression that will mollify him or her* is an optimization problem that blurs the boundary between the cognitive and affective components of problem solving in the same way in which the problem *P: Balance this tray of containers containing hot liquids* blurs the boundary between the cognitive or representational and the behavioral-embodied components of problem solving.

2.2.6.5.1. It is hardly a simple matter to reconstruct and input and output language, a search space of possible or feasible solutions and a verification procedure. But key to both inputs and outputs are a set of descriptors of the set of internal states that trigger different possible external state – contractions of every possible subset of the *33* facial muscles – coupled to a function that maps each possible facial expression – or, each of 2^{33}*-1* possible combination of facial muscle contractions assume each muscle is either *on* or *off*) to an expected response from the person to which *him* or *her* refers. (A full representation of the problem can be built from these primitive descriptors). (It may be, of course, that not combinations of facial muscle contractions are accessible or controllable from the current set of internal states, and that not all internal states that can function as levers of the facial muscle contractions are observable from the current set of internal states. This will make the optimization problem unsolvable in some variable regimes, but not in general.)

2.2.6.5.2. In this case even enumerating the solution space is prohibitively difficult, and, in fact, *attempting* to enumerate the solution space ('in one's mind') will likely lead to failure to produce the optimum or even an admissible action pattern, as it will cause the person trying to do it to produce a facial expression that is counterproductive of his or her purpose.

2.2.6.6. Thus Peirce: 'Inquiry begins with an emotion ('doubt') and ends in an action (predicated on the belief generated by the inquiry)'. Parse 'inquiry' as 'the intelligent deployment of mental activity to the settlement of doubt'. The example above illustrates that perception and action are part of inquiry so defined. The distinction between thinking intelligently and acting intelligently is blurred, as is the distinction between feeling intelligently and thinking intelligently.

2.2.6.6.1. They are all 'mental' – whether they have to do with a representational ('thinking', 'perception of *x* as *y*') or procedural (behavioral, symbolic manipulation of *y*-type structures) process.

2.2.6.6.2. The intelligent voluntary or reflexive movement of the body is no less an example of 'mental behavior' than is the calculation of the Nash Equilibrium for a *2x2* game 'in one's head'.

2.2.6.6.3. Thus Wittgenstein - in his Tractarian embodiment: 'The limits of my language are the limits of my world'.

2.2.6.6.3.1. Query: What is the 'language' of a dancer or a prestidigitator?

2.2.6.6.3.2. Admonishment: One can have 'intelligence without representation' [Brooks, 1991], not only as a matter of building robots based on autonomous processors that control kinematic effectors, but also as a matter of making sense of human patterns of mental behavior.

2.2.7. We are still missing an objective function that mental behavior plausibly optimizes. Taking a line from Peirce and turning it into a question, it makes sense to ask: What is the 'upshot of mental behavior'? – and to try to answer it in the context of the language of problems-languages-solution search spaces-algorithms.

2.2.7.1. Unlike the behavior of inert objects, which, at classical space-time scales can be understood as minimizing free energy (by converting it into kinetic energy: think of a falling mass), mental behavior is difficult to capture in an optimization framework because we need both a global metric ('solving the problem by finding a local or global optimum of the solution space') and a local metric ('do this now if you want to get there then').

2.2.7.1.1. An algorithm gives us a solution concept for this quandary, but not all mental behavior is (purely) algorithmic.

2.2.7.1.1.1. 'Stepping outside the algorithm' to consider whether or not it is the right algorithm for the problem at hand is an important part of any human problem solving procedure. The optimization of *what* is *that*?

2.2.7.2. An operation (prescribed by an algorithm, which is a sequence of operations) is more 'basic', in the sense of 'more granular' – provided that we have the right set of operations to model 'mental behavior' with in the first place.

2.2.7.2.1.1. Like algorithms, operations 'take time' and 'require effort' to perform.

2.2.7.2.1.2. Like algorithms, they map current states of an entity ('mind') into accessible steps ('adjacent steps') with the link provided by the operation.

2.2.7.2.1.3. 'Add 1 to n (and store the result as $n+1$)' is an example of an operation under its relevant aspects: it links two states of the problem solver's mind, it operates 'self-evidently' (although that is because the application rule r is self-evident) on the existing state of K, and it provides a 'reversible' link between the current and the successor states of K.

2.2.7.2.1.4. The upshot of a (computational) operation is analogous to the upshot of a measurement performed on some unknown quantity x. Let a measurement of x be represented by the registration of an interval y_1, y_2 such that $y_1<x<y_2$, and a successive measurement of the quantity x be represented by the registration of a second interval y_1, y_3 such that $y_1<x<y_3<y_2$. Then, the information gain of having performed the second measurement after the first measurement is $I_{12}=\log_2(|y_2-y_1|/|y_3-y_1|)$. Analogously, let x represent the (point) solution to an optimization problem P_O, and u_1, u_2 *and* u_3 represent successive approximations to s from above and below, produced by an algorithm A with the right 'alternating uniform convergence' property. Then the information gain produced by the operation that takes the estimate of s from u_2 to u_3 is $I_{23}=\log_2(|u_3-u_1|/|u_2-u_1|)$, which can be reduced to $I_{23}=\log_2(|u_3|/|u_2|)$, by setting u_1 to 0 without any loss of generality (it is simply a translation along the s axis).

2.2.7.2.1.5. For instance, Newton's (recursive) algorithm for computing the square root of 2 based on successive estimates produces successive estimates to the *5*-digit value of *sqrt(2)* of *1.5000, 1.4167, 1.4142*, … yielding *2* new bits of information per iteration. Since each iteration contains *4* elementary operations, we can calculate the per-operation 'upshot' – the informational gain – as *2/4 =0.5* bits.

2.2.7.2.1.6. Aside: This is a precise result, but, the example on which it is based has some very special properties: an algorithm that chains together operations that produce this marginal informational gain is known – both to us and to the problem solver; the algorithm has a very special alternating convergence property, i.e. successive estimates of the solution approach the 'true value' alternately from above and below – which allows us to use a simple formula for computing the informational gain; and the existence and uniqueness of the answer are both known – both to the problem solver and to the observer or modeller.

2.2.7.3. The informational benefit of a (single mental) operation need not rely on specific knowledge of the topology of the search space, of the precise form of the solution search procedure – the algorithm - and of the dynamics of the convergence of the recursive outputs of the algorithm. A more general approach to representing 'what the problem solver's mind does' is to model its instantaneous state at any place before or during attempting to solve a problem *P* via a probability distribution function *p(x/K:(P;A))* over the value of the solution *x*, conditional upon the state of knowledge *K* of the problem solver – which includes knowledge of the problem *P* and of one or potentially more algorithms {*A*} for solving *P*. The (subjective) 'state of fog' of the problem solver *vis a vis* the value of *x* can be represented by the conditional entropy *H(p(x |K)* which is given by the expected value of the information contained in *p(.),* i.e. by $H(p(x\,|K) = -\int_{-\infty}^{\infty}(p(x\,|K))log_2 p(x\,|\,K)dx$. This conditional entropy models the extent of the problem solver's uncertainty about the true value of the solution *x*.

2.2.7.4. Useful information generated by the application of some algorithm A to the user's existing knowledge K should decrease the conditional entropy H of the problem solver. On a per-operation basis we can measure the net difference in H, ΔH_n that results from the nth mental operation as the number of bits of useful information, I_n, that are generated by it: $I_n = -\Delta H_n$. Useful information decreases the problem solver's subjective uncertainty regarding the specific value of the solution to P. Since entropy is a measure of average information – in this case, the information embedded in the probability that any x is the true value of the solution, x_s, or, $-log_2 (p(x=x_s))$ and information is a measure of the problem solver's 'surprise' at finding $x=x_s$, the conditional entropy measure $H(p(x|K))$ is a measure of the 'expected value of the surprise' for the problem solver at any one point in time, and the decrement ΔH_n in $H(p(x|K))$ due to useful information I_n is the decrease in the expected value of this surprise.

2.2.7.4.1. The question, "How is the problem solver supposed to know how surprised she will be by the discovery that the solution is x_s?" is therefore equivalent to, "How uncertain is the user about the fact that $x=x_s$?", conditional on her knowledge K, which includes the problem P and the specific algorithm A.

2.2.7.4.2. The problem solver's subjective uncertainty regarding the value of x may rise (or fall) in a way that does not track the fall (or rise) of $H(p(x|K))$ that is merely due to the application of algorithm A to problem P to generate I_n. A (human) problem solver may become anxious, or confused, or may make an error in the execution of an instruction that is part of A. The model 'allows' for 'failures of will and wit' or for lapses of memory or concentration: it requires neither perfect recall nor perfect self-control in the deployment of mental energy into the process by which a problem is solved, and in fact it can be used to track such failures. By connecting internal states ('anxiety') to the subjective estimate of the spread of possible solutions $H(p(x|K))$ the model provides a means by which behavioral measures (betting odds on different values of x, which can be used to derive $H(p(x|K))$ at different points of the problem solving process $(n, n+1, n+2)$ can be compared to the increase or decrease in $H(p(x|K))$ which an operation $(n, n+1, n+2)$ *should* provide - i.e. I_n, I_{n+1}, I_{n+2}.

2.2.7.4.2.1. Aside: And is this not what a model should do, i.e. 'to provide means to do something? Think of the Bohr-Rutherford model of the atom: it provides 'means of interacting with atoms' (via excitation patterns). That is *not* what the Ehrenfest model of the electron provides – which may be why you are familiar with the first and not with the second.

2.2.7.4.3. The *cost* of an operation – or, of an ensemble of operations that jointly constitute a 'step' or an iteration of an algorithm A for solving problem P has at least two components: a (possibly physiological) marginal cost associated with the production of new information through an energy-consuming process, c_m, and a working memory cost c_M associated with holding – 'before one's mind's eye' – the quantities that are relevant to the operation (which include the specific instruction associated with the operation, the rules related to the implementation of the instruction, and the input information required for the instruction to successfully execute). The net benefit of an operation n for a problem solver can be written as $U_n = -\Delta H_n - c_m[n] - c_M[n] = I_n - c_m[n] - c_M[n]$. If we sum over n – the number of operations required to calculate the solution to P via A through a sequence of N operations ($n=1,\ldots, N$) to tolerance ϵ, i.e. to stop at x_n such that $|x_n - x_s| \leq \epsilon$, we arrive at the net benefit of using the algorithm A – a linked series of operations that have total net benefit $\sum_{n=1}^{N} U_n(A, P|K)$ to solve problem P on the basis of prior knowledge K.

2.2.7.4.3.1. The problem solver can estimate her 'expected surprise' before she experiences any opportunity to be surprised; and therefore that the local net benefit of a calculation is itself computable. But, in order for the problem solver to solve the decision problem of whether or not to *try* to solve P via A, she must have some *a priori* estimate of N – the maximum or even expected number of operations that will take her to within an acceptable distance from x_s.

2.2.7.4.3.2. But, how would the problem solver – or he who observes and models her – know or form some estimate of N? And, how would a model of the mental behaviour we call 'problem

solving'; accommodate the process by which one might come to know *N*?

2.2.7.5. To estimate *N* in advance of solving a problem, we need a complexity or difficulty measure for problem solving processes that is transportable across problems, across algorithms for solving them, and across hardware or machines on which such algorithms would run.

2.2.7.6. The time complexity $C_T(A|P; K)$ of an algorithm *A* – the number of operations it requires to calculate the solution to *P* via *A* by making use of knowledge structure *K* - is well-matched to the basic ontology of a model of problem solving that includes algorithms, operations, problems and solution threshold criteria. However, the measure is dependent on what is meant by an operation, what the input and output languages in which a problem is defined is, and the degree of generality with which the problem can be 'solved' on different (hardware) instantiations. A general model for a computational device is required.

2.2.7.7. A Turing Machine is a general embodiment of a computational device that can therefore be used to provide a reference embodiment for measuring time complexity.

2.2.7.7.1. It is general in the sense of being universal: If *F(n)* is computable, then it is Turing-Computable (Church-Turing Thesis). This makes it possible to speak about the time complexity of algorithm *A* for solving problem *P* directly in terms of the number of operations required by a Turing Machine embodying *A* to halt;

2.2.7.7.2. It is general in the sense of being a universal 'simulator'. It can be used to simulate the workings of any other digital computational device. Being able to use a Turing Machine to simulate the workings of any other device that implements *A* to solve *P* entails that the complexity of a Turing-implementation of *A* will differ from that of an X-machine implementation by a constant or log-constant, which represents the complexity of simulating the operations of *X* on the TM.

2.2.7.8. Solution algorithms for a problem *P* can be classified in terms of their time complexity $C_T(A|P;K)$ measured on the basis of their implementation on a reference 'hardware platform'. This gives a measure of 'how long it will take to solve *this* problem using *this algorithm*, but not a measure of how long it will take to solve any problem *like this one*. In order to have a measure for the difficulty of solving 'a problem like this', we need a precise grasp of what 'this' refers to: is there a level of abstraction in thinking about problems that allows us to differentiate among problems in a way that is relevant to the difficulty of solving them?

2.2.7.8.1. What is required is a way of parsing problems and algorithms in terms of the relative growth of their complexity with the number of free variables of the problem statement, and of the 'form of the problem statement'. Such a measure would allow us to distinguish between different classes of algorithms in terms of their time complexity.

2.2.7.8.2. A 'polynomial hierarchy' of time-complexity regimes (*Poly, NPoly, etc.)* achieves this 'algorithm sorting' function by distinguishing between classes of functions that map the dependence of $C_T(A|P;K)$ on the number of input or free variables of the problem statement: 'Polynomial-time algorithms' (Poly) halt in a number of operations that is at most (in the worst case) a polynomial function of the number of input variables. 'Non-polynomial-time algorithms' (NonPoly) may require a number of operations that is a higher-than-any-polynomial' (eg: exponential) function of the number of variables to halt.

2.2.7.8.3. We want, however, not only to sort algorithms by the difficulty of implementing them, but to also sort problems by some measure of their difficulty – and expected cost to the problem solver – independently of the algorithm used to solve them.

2.2.7.9. Problems can be time-complexity-sorted by applying the polynomial hierarchy measure we applied to algorithms to the worst-case complexity of solving a problem using any algorithm.

2.2.7.9.1. The Poly-NPoly distinction which separated algorithms according to the functional form of the dependence of $C_T(A|P;K)$ on the number of variables of P becomes the familiar P-NP distinction. P-hard problems have a worst case complexity $C(P|K)$ that is upper-bounded by a quantity that is at most a polynomial function of the number of free variables. NP-hard problems can only be solved in higher-than-polynomial time by a deterministic algorithm (one-head deterministic Turing Machine), but can be solved in polynomial time by a non-deterministic algorithm or a non-deterministic, multi-head Turing Machine (containing at least one 'guess' operation).

2.2.7.10. $C(A|P;K)$ and $C(P|K)$ give us estimates for the number of operations N that are required to solve problem P, which depend either on the general form of the problem and of an algorithm for solving it, or only on the general form of the problem. They do not, however, depend on the specific instantiation of the problem.

2.2.7.10.1. Since each general-form problem – like *KNAPSACK* – has an infinite number of specific instances, $C(P|K)$ equips the problem solver with a complexity metric that is both broadly applicable and computable in advance of beginning to solve P.

2.2.7.10.2. The problem solver can estimate her worst case cost of solving a problem, provided she knows the complexity class of the problem, and the number of variables in the instantiation of the problem she is solving.

2.2.7.10.3. The problem solver can therefore calculate the net (worst case) benefit of solving a problem using only the form of the problem, knowledge of its complexity class, and knowledge of the benefit of solving the problem within a certain time window.

2.2.7.11. Problems that cost a problem solver using a deterministic algorithm 'too many operations' can nevertheless yield to algorithms that make guesses and truncations. Not all guesses and truncations are created equal: some approximation or randomization schemes are better than others: they have better average case cost measures. Average case complexity depends therefore both on the form of the problem and on the form of the algorithm used to solve it.

2.2.7.12. A problem solver can be represented thus: a bundle of (decision or optimization) problems $\{P_l\}$ which she uses to represent 'situations' or predicaments; for each P_l, a set of algorithms $\{A_{lm}\}$ – or solution search procedures that search the solution space of P_l exhaustively, approximately, or randomly – and are deployed to solve P_l as it is instantiated in the problem solver's life to an acceptable tolerance (some are approximate) with an acceptable reliability (some are random); a set of 'cues' or 'frames' $F: D \rightarrow \{SP(\{P\})\}$ that map raw sensory perceptions $\{D_n\}$ onto some subset SP of $\{P_l\}$ and which represent specific adaptations of $\{P_l\}$ to the problem solver's predicament; and a set of worst-case and average-case complexity measures $\{C(\{P_l\}|K)\}$ that determine the problem solver's estimate of the worst- and average-case cost of solving the problem. The solver chooses the best algorithm from among $\{A_{lm}\}$ for solving P_l within SP: the algorithm that maximizes the value of a solution (accuracy and reliability) net of the cost of implementation (based on an estimate of the worst- or average-case complexity of the problem P_l). The problem solver also chooses whether or not to continue to use algorithm A_{kl} to solve problem P_l on an operation by operation basis, on the basis of maximizing the net benefit (informational benefit net of computational cost) of the next operation.

2.2.7.12.1. This model introduces objective functions that guide both high level (choices among algorithms) and low level (stop/start rules at the level of operations) mental behaviour. It is possible – and the model allows for imperfect recall – that the problem solver 'forgets' a high-level decision when making a low level decision and (sub-optimally) abandons the implementation of an algorithm A_{lk} to solve problem P_L

2.2.7.12.2. These objective functions guide mental behaviour in the same way in which an objective or utility function guides physical (choice) behaviour in a micro-economic model.

2.2.7.12.3. These objective functions therefore guide the process by which an agent that (supposedly) maximizes an objective function in fact does so (i.e. they guide the process by which the agent optimizes).

2.2.7.12.3.1. They do so, however, in a way that is locally computable: we do not need top specify a set of objective functions that guide the process by which these objective functions are optimized. They are therefore good candidates for representing the 'upshot of mental behavior' at both a problem-algorithm and algorithm-operation level.

2.2.7.13. The model of 2.2.7.12 relies on three different moves, each of which is problematic:

2.2.7.13.1. the use of a mental 'operation' as a basic unit of mental behaviour:

2.2.7.13.1.1. the apparent scientificity of the word 'operation' belies the imprecision of the phrase 'mental operation'. Are 'adding two numbers', 'tilting the three dimensional image of an oblate spheroid ("in one's mind")', 'recalling the precise location of a compact disk on a large shelf' and 'disambiguating the imprecise use of the word 'operation" all operations? *Mental* operations? What is the sequence of mental operations that 'solves the frame problem'?

2.2.7.13.1.2. Are mental operations to be understood as 'reducible to binary operations that can be implemented on a discrete state random access machine'? 'Adding two integers' is likely to have a binary implementation that is 'intuitive': a real problem solver untrained in the construction of machine code is likely to be able to 'follow' the set of instructions that translates the operation 'add two numbers' into the random access machine

intelligible instructions that produce a binary string as the output. However, the specific implementation of operations such as 'tilting this three dimensional oblate spheroid in your head' on a digital machine is not likely to be intuitive or intelligible to the human that is trying to perform the operation.

2.2.7.13.1.2.1. Does this matter? The fact that 'stereoscopic vision' has a neural implementation that is not intelligible to the viewer (in the sense that the viewer cannot come to 'see stereoscopically' as a result of reading and being able to solve the equations of motion of the neural circuitry comprising the visual cortex) does not make the phenomenon of vision any less mental.

2.2.7.13.1.2.2. The fact that stereoscopic vision is mental in spite of the fact that the neural dynamics underlying it are not intelligible is not relevant. The specific neural sequence of events underlying stereoscopic vision are not 'mental operations'. They may be correlates of mental operations, or that upon which mental operations supervene. 'Tilting (mentally) an oblate spheroid' is a mental operation, whereas '001101010010101010…01010' – the sequence of binary instructions that compiles on a digital machine to an executable file that performs the tilt is not a mental operation, or a sequence thereof. The sequence does not compile on brains – but on digital devices. Moreover, the mental operations that correspond to reading and making sense of the sequence of digits 001101010010101010…01010' are completely unlike the mental operation of tilting the image of an oblate spheroid.

2.2.7.13.2. the use of a Turing model for the computational processes that track or represent mental behaviour:

2.2.7.13.2.1. a Turing machine is a device built for universality, not for intuitiveness or plausibility. It is meant to simulate any other

discrete state computational device and to mimic, via minimal and idealized hardware, the process by which an algorithm 'works', not to provide the most intuitive implementation of a very particular set of algorithms (those associated with the 'everyday life' of a real mind-brain). Computational complexity costs associated with Turing machine implementations of algorithms can only indicate the difficulty of a problem in a worst-case instance of a particular case.

2.2.7.13.2.2. If it were the case that real humans rely on a very special and narrow set of algorithms which have received optimized (neural) implementations (i.e. 'brains'), then Turing Machine or Random Access machine models of algorithmic procedures that represent mental behaviour would 'miss the point' of why certain mental operations are more or less costly and more or less beneficial than others, because it is only in the context of a particular neural implementation that this question has a definite answer.

2.2.7.13.2.3. Turing introduced the discrete state one tape machine as a model of mental operations on the basis of a set of paradigmatic examples (addition, matrix multiplication and inversion, the algorithmic deployment of truth tables to determine the truth value of well-formed formulas of first order logic) of problems and operations that relate to symbolic manipulation. Its success at providing a model of generalized computation rests on the very large scale on which such problems are considered paradigmatic of mental functioning. But, paradigmaticity is not the issue here; 'coverage' is. 'Implementing' – on a TM-simulable RAM - the mental operation(s) that represent 'purging a stanza of its sarcastic overtones' should give pause to a RAM programmer.

2.2.7.13.3. the privileged use of mental or cognitive problems to represent 'that which human agents do when they solve problems'.

2.2.7.13.3.1. 'Persuading *X* (by time *t*) to do *Y* (by time *T)*'; 'balancing a (full teacup) on my head for at least 54 seconds'; 'arpeggiating (on a keyboard) a six-octave diminished seventh chord starting on c(4) sharp in a fulsome *fortissimo* tone' are all problems that require a combination of representational ('cognitive') and behavioural ('action') processes (or, 'operations'). If the essence of a problem if 'getting there (desired state) from here (current state)' then all of these are problems in precisely the same way as *"22878975 x 98709870008 = ?"* But, they are not problems in the sense in which their solution algorithms are transparently implemented on a TM.

2.2.7.13.3.2. The model 2.2.7.12 replaces 'decision agents' with 'problem solving agents' – or, problem solvers - for the purpose of unpacking processes ('optimization', 'decision') that are nebulous to economists and decision theorists - and their uninformed readers - but in doing so makes use of the concept of a 'problem' that does not allow for the co-extensive nature of sensing, perceiving, feeling inferring, learning, optimizing and behaving that we would expect of a model of real humans.

2.2.7.14. We are here after 'intelligent artificiality', not artificial intelligence.

2.2.7.14.1. We are not after getting a digital device to mimic the (usually verbal, in practice, even though grander claims are usually made 'in theory') behavior of a human. That is the problem of AI – the problem Turing articulated.

2.2.7.14.2. We are after a model of mind-brain process, procedure and performance that we can use to interact with real mind-brains and the behavior they produce: we can predict and control it.

2.2.7.14.2.1. Just like certain models (of electrons) give us good reason to think electrons are real ('because we can spray them') because they provide for means (levers) by which we can make electrons do things, the models that together comprise 'intelligent artificiality' should allow us to interact with minds-brains ('so we can change them' – and NOT merely 'so we can explain them' and 'so we can simulate them'.)

2.2.7.14.2.2. 'Controlling X's behavior' is different from 'predicting X's behavior' in at least two important ways:

2.2.7.14.2.2.1. It requires predicting changes in the value of some variable(s) causally relevant to X's behavior with a pre-set period of time (the 'action window'). 'Late predictions' are useless to the controller.

2.2.7.14.2.2.1.1. And, 'predicting' in this case is not in any sense equivalent to 'explaining', as (some) economists (seem to) believe. Prediction is not 'nothing but explanation in reverse', even if the person producing the explanation is 'blind to the data' in the sense of not having actually 'seen the raw data'. In 'producing an explanatory model' the explanation-producer knows the nature of the variables the values recorded are values of – and therefore already knows the basic 'ontology', 'output state space' or 'chema' that will be used to encode the data. This information will always only be *retro*dictively and not *pre*dictively available. By contrast, prediction relates to a situation where this knowledge is not available at the time a model is articulated. The 'fog of the future' relates not only to the value of the variables but also to the nature of the variables – the identity, numerosity and topology of the state space of the system whose behavior one wants to predict. This distinction is what separates (most of the) social sciences as they are now practiced from the sciences that produce *techne* and *phronesis,* which is what intelligent artificiality aims to produce.

2.2.7.14.2.2.2. It requires intelligent, adaptive, causally connected 'action upon X' that is itself time-bound.

2.2.7.14.2.2.2.1. The apparatus of the Millikan and Michelson-Morley experiments are part of the 'models' that guided those experiments. They provide 'physical levers', not just 'mental maps'.

2.2.7.14.2.2.3. It requires real time adaptation to the putative responses of *X* 'as they happen'.

2.2.7.14.2.2.3.1. An adaptive filter deployed in a wireless broadband data engine (encoder/decoder-equalizer, modulator, demodulator) estimates variations in channel conditions 'as they happen' in order to increase the accuracy and reliability with which an incoming signal is decoded. A 'brain state modulator' requires, analogously, real-time estimation of the dynamics of cortical responses in order to select and effect the most efficacious combination of inputs for achieving the purpose of the modulatory task it is designed to implement.

2.2.7.14.2.2.4. It requires real time computation of an optimal or sufficient or melioristic response to the changes in the state(s) of *X*.

2.2.7.14.2.2.4.1. A force-feedback-based control system for an actuator designed to 'balance a tray of containers filled to various degrees with hot liquids' requires real-time computation the distribution of forces produces by the tilt of the tray at different points along its surface, so that it can re-distribute the

values of changes in forces along the surface of the tray 'in time', and with the minimal reliability and accuracy required to produce the intended effect.

2.2.7.14.2.2.5. It requires real time actuation or effectuation of an optimal or sufficient or melioristic response to the changes in the state(s) of *X*.

2.2.7.14.2.2.5.1. The said force-feedback control system not only observes (senses, measures, registers) and computes: it also *does*: i.e. it intervenes in the behavior of *X* in a way that is causally connected to the state variables of *X*.

2.2.7.14.2.2.6. It requires real time prediction of the effect of any actuation scheme on the dynamics of the system upon which it acts.

2.2.7.14.2.2.6.1. Successful force feedback control of complicated objects requires real time estimation of at least the transient response of the system to any pattern of impulses (changes in force distributions) that are aimed at maintaining or restoring 'balance', for instance.

2.2.7.15. Intelligent artificiality is neither a new kind of science nor a new science. It is in fact not a science in the way in which the word usually refers.

2.2.7.15.1. It is a discipline guided by a set of specific set of regulative objectives that serve to shape inquiry, communication and action.

2.2.7.15.2. It is a craft oriented towards the reliable production of precisely specified effects whose collective impact is to solve practical problems ('emotional self-control', the production of a mind-brain-body state of 'participative or empathetic detachment', the selective training and development of a new cortico-motor skill, such as playing the sixteenth-note passage in sixths in the second of Brahms' Paganini Variations Op. 35 *a tempo*.

3. Minds-brains 'optimize' in the course of doing what they do. But:

3.1. What do they optimize?

3.1.1. 'Utility' (eg. pleasure, the inverse of pain) gives no guidance as to that which is optimized by and within the process of optimization.

3.1.1.1. Optimization itself is neither free nor instantaneous, but that which mind-brains always 'do', or, are in the process of doing, not what they always-already have done or will have done by the time we get to representing them.

3.1.2. The 'physiological utility/dis-utility' of mental behavior is a fuzzy concept because it has no normative regulative principle to guide its application.

3.1.2.1. One may experience idiopathic anxiety or 'panic' while attempting to solve a decision problem, which will cause one to leave off the process of solving the problem: an instance of physiological disutility minimization. What useful insight may be inferred from this? The fact that the anxiety is idiopathic leaves open the possibility that the problem solver has rationally (on a cost benefit basis) decided that the procedure by which she was attempting to solve the problem was not sufficiently productive (in terms of bits per operation) for her to anticipate being able to solve the problem within the time window allotted to it. On the other hand, the anxiety may be caused by a(n idiopathic) surge in epinephrine levels, co-instantiated with thoughts about the possible onset of ventricular tachycardia.

3.1.3. The 'informational utility of mental behavior' – on a per-operation-within-algorithm and per-algorithm conditional on a definition of basic mental operation basis works well as a regulative concept because it allows us to represent and track problem-solving-directed sequences of mental events and compute a net upshot of mental behavior. However:

3.1.3.1. It relies on an apparatus for measuring local utility/disutility (entropy, conditional entropy) and for measuring expected costs of sequences of operations (weighted time complexity of worst case instantiation) that is contingent on a Turing machine/Random Access Machine model of mental process that highlights the contingency of both these measures on

the definition of a basic set of operations, and on a(n implausible) capacity for estimating the time complexity of canonical problems.

3.1.3.2. It relies on worst case rather than average case or adaptive-case or adaptive-average-case measure of problem/algorithm time complexity that makes most problems of practical interest (including everyday problems such as 'self-control', force-feedback-based balance recovery and maintenance, ergonomically efficient design of gait kinematics) unlikely to be tackled, let alone solved, by real human problem solvers.

3.2. How do they optimize? *How*… is more difficult to parse than *what*… in this case. It minimally refers to the physical instantiation (neural components and architecture) that are possible and plausible substrates for the optimization functions plausibly ascribed to mental behavior.

3.2.1. 'Digital machines' – whether idealized (Turing Machines) or implementable on custom or merchant silicon (Random Access Machines) impose constraints on architecture (deterministic operations, quantized (rational) connection coefficients among hardware components, non-evolvability of basic architecture) that do not translate to description language that has evolved to describe and manipulate brain structure and function (recurrent neural networks, real-valued synaptic weights, adaptive modification of inter-neuron link weights, adaptive re-distribution of connections ('plasticity')) without losing their most attractive features (universality, universal simulability of other machines in the same class, computability of informational benefits and computational costs of optimization within a unitary framework for modeling the process by which optimization takes place).

3.3. The task of intelligent artificiality is to jointly answer the why-how questions by positing a model that jointly allows for 'representing, observing, computing, and intervening' (in) the real time functioning of mind-brains. To do so it must:

3.3.1. Specify an objective function that mind-brains plausibly optimize at *multiple* time scales, for instance, both at the time of solving the decision problem regarding whether or not to attempt to solve an optimization problem, and durring the process by which the optimization problem is solved; and,

3.3.2. Specify a neurologically plausible architecture that implements the tasks and operations associated with the tasks and operations entailed by:

3.3.2.1. Computing the optimisand (locally and globally); and,

3.3.2.2. Extremizing the optimisand (locally and globally).

3.3.3. The objective of this model is to simultaneously provide both prongs of the constitutive-regulative components of a useful model, via:

3.3.3.1. a set of regulative principles that guide ('regulate') mental behavior and act under the constraints of,

3.3.3.2. a set of entities and principles that 'constitute' what it means for a mental process to be 'embodied' in a brain.

3.3.4. The upshot of this model is to allow for the seamless and self-consistent representation - *qua* embodied mental behavior – of all of what we currently call perceiving, sensing, orienting, emoting, feeling, reasoning, calculating, deliberating and 'acting' or behaving – as part of a micro/macroscopically guided extremization process that 'works' on real brains at different time scales, and to enable a new set of techniques and technologies aimed at predicting and controlling mind-brain states (as opposed to explaining or pretending to predict them) in real time, using real people in real situations (as opposed to undergraduate psychology students in unrealistic laboratory predicaments).

3.4. A first cut at such a model [Dayan and Hinton, 1995; Friston, 2006; 2007; 2008; 2009; 2010] could proceed as follows:

3.4.1. Let $\bar{x} = (x, x_t, x_{tt}, \dots)$ represent a set of vectors $x_{t(n)} = \frac{\partial^n x(t)}{\partial t^n}, \forall t$ that encode the sensory states (neural encodings of environmental variables) $\{x\}$ and their time derivatives x_t, x_{tt}, $x_{tt}^{(n)}$, … .

3.4.2. Let $\propto = (\propto_1, \propto_2, \dots, \propto_k)$ represent a set of actions that the problem solving agent performs on the environment, whose effects on the environmentally caused variables $\{x\}$ is represented by $p(\bar{x}/\overline{\propto})$.

3.4.3. Let $\theta = (\theta_1, \theta_2, \ldots, \theta_m)$ represent the (unknown) environmental causes of $\{x\}$.

3.4.4. Let $s = (s_1, s_2, \ldots, s_n)$ represent a set of internal states of the agent which shape her inferences about the most likely causes $\theta = (\theta_1, \theta_2, \ldots, \theta_m)$ of $\{x\}$ via a probability density function $q(\theta; s)$.

3.4.5. Then the agent's 'general purpose optimization problem' – both locally computable and instantiated at different time scales - is that of minimizing the free energy associated with the set of sensory states $\{x\}$ and internal states $\{s\}$, *i.e.* $\min \quad F(\bar{x}, \bar{s}/\propto) = -\langle \ln p(x, \theta/\propto \rangle_q + \langle \ln q(\theta, \bar{s}) \rangle_q$.

3.4.5.1. The interpretation of the free energy minimization condition is simple in both informational and thermodynamic terms:

3.4.5.1.1. In informational terms, the agent('s mind) is minimizing the expected value of her surprise $-\langle \ln p(x, \theta/\propto \rangle_q$ at experiencing x conditional upon taking action $\propto$ on the background of a probability density function q which depends on her internal states s - net of the spread or entropy $\langle \ln q(\theta, \bar{s}) \rangle_q$ of q. Call this the informational or 'mental' interpretation of F.

3.4.5.1.2. In thermodynamic terms, the agent is actively minimizing the difference between the energy and entropy of its internal states ('available, or, free energy') though an active process of perception (sampling the environment), inference (minimizing surprise by updating expectations through computational work) and action (acting upon the environment to minimize free energy of its internal states). Call this the physical or 'neurophysiological' interpretation of F.

3.4.6. The objective function F is both constitutive of what mind-brains do and regulative for what they do. It is constitutive in the sense that the optimization process that implements: $\min \quad F(\bar{x}, \bar{s}/\propto) = -\langle \ln p(x, \theta/\propto \rangle_q + \langle \ln q(\theta, \bar{s}) \rangle_q$ embodied in the neurological dynamics underlying 'mental events'. The specific adaptations – perception, inference, representational learning, action that supply the variables of the optimisand and the functional form of their relationships are on the other regulative of mind-brain function.

3.4.6.1. We are in the possession of a generalized 'problem solving framework' that describes mind-brain function by positing a problem that is always-already being solved (at different space-time scales) by the mind-brain. (It remains to be shown that the optimization is 'implementable' on structures that resemble brains in meaningful ways, and that such an implementation allows us to 'do things with brains' that other models – viz 'rational choice theory' do not.)

3.4.6.1.1. We are therefore not dependent on a specific problem form (eg *KNAPSACK*, *kSAT*), generic algorithm *('breadth first search', parallel terraced scan')*, meta-algorithm (*Branch and Bound, Stochastic Hill Climbing*), or specific Turing Machine model (based on a transition function among its states that model a specific algorithm for a specific problem on the basis of specific halting rules and criteria) for modeling 'problem solving agents' in a way that allows us to capitalize on the full pragmatic, regulative and descriptive benefit or upshot of a 'model'.

3.4.6.2. The extremization
$\min \quad F(\bar{x}, \bar{s}/\propto) = -\langle \ln p(x, \theta/\propto) \rangle_q + \langle \ln q(\theta, \bar{s}) \rangle_q$ is not to be understood as 'mental' in the sense of 'cognitive', 'perceptual' or even 'representational'. It is to be understood as a general purpose regulatory ('goal directed', as specified by the referents of the terms of the optimisand') and constitutive (the analytical form of the optimisand) model for mind-brain function, which relates to a continuum of perception, learning and action, i.e.:

ACTION	PERCEPTION	LEARNING
$\min_{\{\propto\}} F$	$\min_{\{s\}} F$	$\min_{\{s\}} F$

3.4.6.2.1. The agent minimizes free energy 'over her actions' by effecting causal changes in the environment that drive the exchange of energy towards the internal state corresponding to the maximum conditional predictability of the situation the brain-body-object system 'finds itself in.

3.4.6.2.1.1. Effecting contractions of muscle spindles in the digits and the forearm that counteract instantaneous changes in forces exerted by a large tray of containers filled with hot liquids on the palm and digits 'balances' the tray by causally shaping the environment (digits+tray+containers+liquids) to maintain the dynamics of the brain-body-object system predictable relative to a model of 'smooth continuous translational motion' of the body.

3.4.6.2.2. The agent minimizes free energy 'over her perceptions' by selecting (which should *not* be interpreted in choice of decision-theoretic terms, but rather in terms of a precise description of a reflexive action, as in 'selecting' the right combination of muscles to contract *while* falling *upon* slipping on a banana peel) a (subset of) her internal states (corresponding to encodings of 'salient' variables vis a vis the vector x of relevant states) that maximize the conditional, weighted predictability of the brain-body-object 'situation'.

3.4.6.2.2.1. One selects the set of stimuli (pressures at different points on the surface of the palm and the tips of the digits) corresponding to the forces effected by the tray full of hot liquids on the palm an digits that provide the optimal estimates for the dynamics of the tray conditional upon actions (muscle spindle effectuations) and the model of smooth translational motion.

3.4.6.2.2.1.1. What will *not* likely happen if and while the tray 'feels it is about to tip over', for instance, is that the agent will focus on an itch occurring at the base of the third digit of her free hand.

3.4.6.2.3. The agent minimizes free energy over her 'causal beliefs' by selecting the set of associative relationships ('if, then, else') that maximize the likelihood of her perception and model-conditional and weighted predictions of the brain-body-object dynamics, contingently upon the actions she takes.

3.4.6.2.3.1. The dynamic, causal model that links the effectuation of specific muscle spindles in the palm, digits and forearm to the restoration of vertical force balance on the tray of containers filled with hot liquids is a set of relationships linking subsets of muscle groups that are 'available for contraction' to subsets of possible and observed stimuli that are indicative of ensuing dynamics, to a set of future states of the brain-body-tray system. The agent chooses the set of relationships (independent/dependent variable sets, functional form, including delays, stochasticity, nonlinearity) that maximize the likelihood of future stimuli conditional on a model of the system based on 'smooth, translational motion'.

3.4.6.3. Aside: Much us is made of a 'predictive model', *infra*, which is regulative of the agent's actions, perceptions and inferences. Ponder what would happen if the agent were a prestidigitator, or a four-handed prestidigitator, who had, with practice, learned to catch a large number of falling objects, quickly balance them, and set them down smoothly, one by one.

3.4.7. Implementability of $\min \quad F(\bar{x}, \bar{s}/\propto) = -\langle \ln p(x, \theta/\propto) \rangle_q + \langle \ln q(\theta, \bar{s}) \rangle_q$ 'on brains' should be thought of in terms analogous to those of 'implementability of an *N*-point Fast Fourier Transform (*FFT*) Algorithm - on a data vector with a clock rate of *C* digitized at *B* bits/sample and with coefficients of the unitary *FFT* matrix quantized to *J* bits of precision - on an *H*-bit digital signal processor with RAM size *L* and effective computational velocity of *G* mega-operations/second, where 'operations' may be 'multiply/complex multiply' or 'multiply-accumulate' operations.

3.4.7.1. In the example 3.4.7 one knows or 'can calculate' clock rates, input rates, output rate and precision requirements (for the output signal to be 'useful'), as well as a set of structural characteristics of the 'hardware' - including an effective computational speed measure that includes 'memory access times', input-output times', random access memory size, and effective precision with which the incoming data, the coefficients of the FFT matrix and the maximum precision with which the output of the overall operation are stored, as well as the information storage properties (sample rate of incoming data, memory required for storage or buffering of the data, required output data rate) of the entire process.

3.4.7.2. One can therefore calculate 'in advance' whether or not a machine with a particular set of performance measures (storage, computational speed, register size for representing data samples and along-the-way intermediate products) will be able to 'implement' an FFT algorithm of particular computational complexity and informational storage load, in a certain amount of time ('real time', for instance, i.e. at the rate at which the data comes in) and with a certain precision (H bit quantized output samples of the frequency-domain signal, for instance).

3.4.7.3. One can also calculate in advance 'where the machine should be' at any one point in the computation process, from knowledge of the algorithm it uses, its register states, the data rate(s) for inputs and outputs, and the specific memory and interface management system used by the machine.

3.4.7.4. One can also intervene in the workings of the machine while it is working and observe the effects of its interventions on its computational performance. For instance, one can speed up/slow down its clock speed and observe its effects on internal memory buffer loads, output rate, and output accuracy; increase or decrease its internal memory allocation and observe the effects of these changes on the instantaneous computational load of the machine, as well as the rate and accuracy of its output; increase or decrease the accuracy (in bits) with which data is represented and observe the effects of these changes on the computational load of the machine and the accuracy of the output; among many other interventions, some of which include subsets of the interventions above.

3.4.7.5. The entities in terms of which these interventions are represented (internal memory, computational load, accuracy of output samples) are real 'because you can spray them': you can do things with and to them in order to control the dynamics of the machine and to make precise point predictions about its behavior.

3.4.7.6. The 'model' of the machine 'works' in the control-predict' sense not only in the 'control-pretend-to-predict-while-explaining' sense.

3.4.7.7. Part of the reason for its success is the fact that the machine is designed, engineered and built (for the purpose of computing unitary transformations of data blocks, for instance), not 'given' or 'found'.

'Memory register' refers to a memory register because the machine has been built in such a way as to make this reference relationship rigid.

3.4.7.8. 'Not so with brains' – one might say: 'we did not design them'. But the qualities of the model 3.4.7 et seq are the right qualities that a model of mind-brains should seek to maximize.

3.4.7.9. The fact that 'representation' and 'explanation' are subservient as regulative norms (for modelers that aspire to the qualities of 3.4.7) to controllability and predictability is what makes intelligent artificiality a radical departure from decision theory, rational choice theory, and various sorts of cognitive science and neuroscience.

3.4.8. The question of implementation of $\min \quad F(\bar{x}, \bar{s}/\propto) = -\langle \ln p(x, \theta/\propto) \rangle_q + \langle \ln q(\theta, \bar{s}) \rangle_q$ 'on brains' is equivalent to the question of providing a set of architectural specifications that are enabling of the computation of the FFT algorithm in the example 3.4.7, given a sample rate, a system clock frequency, and a set of output performance measures (rate, accuracy), given a set of functional building blocks (First-in-first-out registers, memory blocks, etc) and a set of associated structural building blocks (transistors, biased gates, leads) associated with them.

3.4.8.1. Mapping the implementation of the optimisand 3.4.5 onto the hardware architecture of a general purpose digital signal processor proceeds by first creating a finite-precision algorithm for the implementation of the optimization problem subject to assumptions about the probabilities p and q (for instance, via gradient ascent/descent conditional upon functional forms of the two distributions) and then coding the algorithm in a language that maps various algorithmic operations onto electronic state behavior changes of the machine elements of the DSP. Of course, one can only do this transparently if the machine language used to program the DSP is easily mapped into the various operations, routines and sub-routines of the algorithmic implementation of 3.4.5. Otherwise, one will attempt to solve a (provably) intractable problem by blind trial and error.

3.4.8.2. A practical alternative is to map the behaviors of various structural blocks and sub-blocks of the computational device onto particular sub-routines and routines and operations of the algorithmic representation of 3.4.5 (e.g. 'multiplies', 'differences', 'storage', 'call') such that every routine and subroutine in the 'library' used by the algorithm can be represented as the time bounded behavior of one of the structural blocks, eg: 'multiplication', 'comparison', 'discrete convolution', etc. the resulting sub-routines correspond to a 'middle' or 'translation' layer between the high level operations described by the algorithm and the low level ('machine') code which regulates sequences of physical state changes in the machine. The 'translation layer' functions as a rigid connector between the abstract 'symbol manipulation' language of the algorithm and the 'embodied' physical state change sequence sets of the machine.

3.4.8.3. The translation layer therefore functions as a 'language of thought' for the DSP, in the strict sense that its terms ('operations') rigidly refer (by design, an in the sense that specific changes in operations require specific changes in their hardware implementation) to sets and sequences of physical state changes that the machine executes in virtue of 'implementing' the algorithm represented at the upper or most abstract layer (algorithm or pseudo-code).

3.4.9. Implementability of 3.4.7 'on brains' should therefore refer to the correspondence of the operations involved in an algorithm that performs the extremization 3.4.5 and the operations that a neural network with 'cortical' properties can be used to implement; as well as the specific architecture required to implement the extremization 3.4.5. and the architecture of such a neural network. It should show how a 'brain-like structure' can implement solution search procedures for problems whose 'Turing-computable' solution search procedures belong to 'intractable' time complexity classes', for realistic numbers of independent problem variables.

3.4.9.1. Cortico-cortical connections in the histologically distinct parts of the brain (eg. Visual Cortex) that (putatively) 'implement' entropy-biased weighted minimization of the Kullback-Leibler distance between a set of predictive beliefs p (or, perceptual expectations) and a set of recognition densities q are partitioned into a set of layers of neurons that are connected in forward and backward directions with different topological features (forward connections are more topographically organized) and different degrees of connectivity (there are more backward connections

than forward connections) and bifurcation (there are more backward-pointing bifurcations than there are forward-pointing bifurcations).

3.4.9.2. A recurrent neural network architecture with both forward and backward connections – the Helmholtz Machine [Dayan and Hinton, 1995] – can be used to implement a hierarchical form of a Bayesian inferential process of causes from inputs [Friston, 2003; 2010] that is not subject to the exponential expansion of the solution search space (of possible causes for a set of sensory inputs) that is a well-known drawback of maximum likelihood algorithms [Dempster, Laird and Rubin, 1976].

3.4.9.2.1. It does so by replacing the problem of predicting inputs from causes with the problem of inferring causes from inputs, relative to a recognition density over the causes and associated internal states, i.e. $q(\theta, s)$.

3.4.9.2.2. The (potentially intractable) inference problem of deducing 'the best explanation for a set of sensory inputs' is rendered tractable by a multi-layer encoding structure that implements a set of simple error computations between expected and inferred inputs (to each layer of neurons) on the basis of a set of probabilities defined over possible causes.

3.4.9.2.3. The range of adaptations of the resulting problem solving agent (who now uses a brain-like structure to make inferences. Recognize objects and act on the environment) may be decomposable according to the time scale on which these adaptations take place, i.e.

$$q(\theta; s) = \prod q(\theta_i; S_i) = q(\theta_p; S_p) q(\theta_m; S_m) q(\theta_M; S_M)$$

where the subscript 'p' denotes 'pico-state changes' (neuronal connection strength changes, occurring at the level of hundreds of milliseconds); the subscript 'm' denotes 'micro state changes' which can refer to attentional changes operating on time scales of seconds, and the subscript 'M' denotes macro-state changes, operating on longer time scales, and may refer to plastic cortico-cortical connectivity changes.

3.4.10. The complexity of the model is not in the complexity of its implementation.

3.4.10.1. It is not the ability to perform faster sequential computations that distinguishes agents of different 'power'.

3.4.10.1.1. That is an image of 'intelligence' that has drawn its strength from the appeal of the Turing Machine as a general model for thought, but it needs to be modified.

3.4.10.2. Computational prowess – in the sense of being able to solve optimization problems of a known maximal worst case time complexity in a given period of time – is not the right metric for the problem solving prowess of an agent.

3.4.10.2.1. That is because 'the hardware' restricts both the upshot and the bounds of computational speed in the extremization of 3.4.5. 'By design' - one would like to say, if only it did not trigger irrelevant associations.

3.5. The complexity that matters to distinguishing problem solving agents and processes is informational, not computational.

3.5.1. The algorithmic or Kolmogorov complexity *Kolm(x)* of a string *x* (a binary representation of a representational object) is the length of the minimal program that reconstructs or synthesizes that string.

3.5.1.1. 'Reconstructs that string': How?

3.5.1.1.1. As the output of a computational device that runs the program whose length is the complexity measure.

3.5.1.2. 'Reconstructs that string': from what?

3.5.1.2.1. From a minimal set of inputs that form part of the program whose length is the complexity measure.

3.5.1.3. 'Reconstructs that string': How quickly?

3.5.1.3.1. In a finite number of operations (which could be very large, of course, and may scale very quickly with the state space of the reconstruction problem – for instance, it may contain many recursive loops).

3.5.2. Let $Kolm_M(x)=l_{min}(A(\boldsymbol{x}))$, represent the Kolmogorov complexity of a (set of strings) vectors $\boldsymbol{x}$ with respect to machine M. (I will drop the boldface henceforth, and keep x to refer to the set of strings and $x_{subscript}$ to refer to one of them. Consider an agent that is interested in an encoding schema for x that will allow him or her to most efficiently process x-relevant information, for instance, to process a sequence y in order to decide whether or not y and x are identical (not in a Hamming-distance sense, which would be a trivial case of *xOR-accumulate* operations but in the sense of having been generated by the same algorithm working on a similar machine, even if it is working on a different set of inputs (generalized deconvolution). Then the agent would encode the strings x_k in a way that assigned probabilities of occurrence of strings with greater complexity that are lower than those of strings with lower complexity.

3.5.2.1. Like natural language (viz. Zipf): shorter words occur more frequently across all human-produced texts and speech acts.

3.5.2.2. Like block-channel-coded digital communications systems: code words with lower probabilities of emission by a coding source have greater (incompressible) lengths.

3.5.2.3. Like block-source coded digital compression systems: smaller blocks occur more frequently.

3.5.2.4. Kraft-McMillan coding implements length-dependent signaling by assigning probabilities to strings x_k on the basis of their length $l(x_k)$, eg: $p(x_k)=2^{-l(x_k)}$.

3.5.2.5. Suppose the encoding of the x_k were related to the *fundamental* length of x_k in a way that mimics the operation of an efficient coder/decoder. Then it makes sense to replace $l(x_k)$ with $Kolm(x_k)$ in the ascription of probabilities to meaningful strings, i.e. $p(x_k)=2^{-Kolm(x_k)}$. A complexity-aware encoding scheme for probabilities (or conditional probabilities) of sensory inputs (upon causes, say) or causes (upon internal states) would

proceed by assigning probabilities in increasing order to strings on the basis of the (inverse of) their algorithmic complexity. The entropy of the resulting state space is then simply the average Kolmogorov Complexity of the strings representing the states over which the probabilities are defined:

$$H(x) = -\sum_{k=0}^{N} p(x_k)\, log_2\big(p(x)\big)$$

$$= -\sum_{k=0}^{N} 2^{-Kolm(x_k)}\, log_2\big(2^{-Kolm(x_k)}\big) = \sum_{k=0}^{N} p(x_k)\, Kolm(x_k)$$
$$= Kolm_{ave}(x_k).$$

3.5.2.5.1. *Kolm(.)* is defined relative to a computational device *M* (a Turing Machine serves the purpose well because of its universality), i.e. $Kolm(.)=_{def}Kolm_M(.)$ and its universality is guaranteed by the Invariance Theorem that states that the Kolmogorov Complexity of a string defined relative to a machine can differ by no more than a constant from the Kolmogorov Complexity of the same string defined on any other machine, i.e. $Kolm_M(x)=Kolm_N(x)+ O(c)$.

3.5.2.5.1.1. However, a (large) constant can be very meaningful when the agent has *10s* to solve a problem of causal inference from sensory inputs and priors over putative causes.

3.5.2.5.1.2. Invariance comes at an implementation cost that a brain-constrained agent may (and often will) not afford.

3.5.2.5.1.3. The choice of the machine relative to which *Kolm(.)* is calculated matters to the relevance of *Kolm*(.) to the implementability and implementation of the minimization 3.4.5.

3.5.2.5.1.4. Fortunately the universality of *Kolm(.)* allows us to restrict its definition to the class of recurrent neural networks with properties satisfying 3.4.9.1.

3.5.2.6. $Kolm_M(x)$ should be defined in terms of the algorithmic content of x (the length of the shortest program that produces x as its output) on a machine that has the architectural and operational structure of a recurrent neural network that tracks experimental evidence on cortico-cortical connections, $Kolm_{CORT}(x)$.

3.5.2.6.1.1. The restriction has immediate repercussions for the computational work that may be required to either synthesize $Kolm_{CORT}(x)$ (search among finite length programs implementable on CORT, say, which is exponential in the length of x in the worst case) or to synthesize $p(x)$ from $Kolm_{CORT}(x)$ (production of a *Kolm*-coded distribution function satisfying the probability axioms).

3.5.2.6.1.2. 'No computation without implementation': $Kolm_{CORT}(x)$ needs to be replaced by a resource-bounded variant, which restricts the complexity class of the problems that the machine CORT needs to solve in order to synthesize *Kolm* from x and $p(x)$ from *Kolm(x)* (to the classes LogPoly or LINEAR.).

3.5.2.7. The computational power of CORT does not reside in what we typically think of as 'computation' (following Turing): the sequential implementation of rapid sequences of *XOR, AND, NOT*-type operations. It resides in the informational complexity *Kolm* associated with the state spaces of its generative *(p)* and recognition *(q)* probability density functions.

3.5.2.7.1. It is a function of the informational capacity of the network subject to computational complexity (resource) bounds.

3.5.2.7.2. It is encoded by (real-numbered, 'analogue') weights of cortico-cortical connections, and therefore it is not a measure of the logical depth of state-dependent cortical processing.

3.5.2.7.2.1. Humans 'differ' most significantly in fundamental informational (*Kolm*) breadth, not in computational *(Time(A(x))* depth.