Perhaps the Rigorous Modeling of Economic Phenomena Requires Hypercomputation ⋆

SELMER BRINGSJORD[1†], NAVEEN SUNDAR G.[1‡], EUGENE EBERBACH[2¶], YINGRUI YANG[1§]

[1] *Dept. of Cognitive Science, Dept. of Computer Science, Lally School of Management (1 only)*
Rensselaer Polytechnic Institute (RPI), Troy NY 12180-3590
[2] *Department of Engineering and Science, Rensselaer Polytechnic Institute (RPI), Hartford CT 06129-2991*

received 6 April 2010; in final form 28 September 2010

CONTENTS

LIST OF FIGURES

LIST OF TABLES

⋆ This paper is partially based on a paper presented (by Sundar G.) at the Hypernet 10 workshop, Tokyo 2010.

† email: selmer@rpi.edu
‡ email: govinn@rpi.edu
¶ email: eberbe@rpi.edu
§ email: yangyri@rpi.edu

1 INTRODUCTION

Mathematical logic was born from the rigorous formalization and study of mathematics by way of computation and formal logic, and includes such seminal discoveries as Gödel's incompleteness theorems and Turing's halting-problem result, both proved at the dawn of computer science. These discoveries revealed that a core problem in mathematics (viz., deciding whether or not a sufficiently rich proposition is a theorem, and producing a proof that certifies the answer), in certain contexts, is *tremendously* difficult. Mathematical logic continues to be quite fertile; recent results include demonstrations that certain theorems in mathematics are in the "independent-of-Peano-Arithmetic" class drawn to our attention by Gödel, (e.g., Goodstein's Theorem; for a nice overview, see [40]), and are so hard to obtain that humans currently find it necessary to deploy infinitary concepts and reasoning in order to secure them. Parallel comments could of course be made about other parts of formal logic, for example, the economics-relevant part pertaining to the modeling of dynamically interacting agents with beliefs, perceptual abilities, goals, and so on (e.g., see the system — heavily indebted to preceding agent-oriented logical systems — presented in [1]).

Might it be that the rigorous formalization and study of economics, carried out from the perspective of formal logic and computability theory, will reveal that core problems in that field are likewise tremendously difficult? Specifically, might it be that some economic phenomena involve hypercomputation, or are at least on plausible frameworks for doing science best modeled using schemes that are in part hypercomputational? In this short paper we defend an affirmative answer to this question.

Please note that even if our answer is correct and successfully defended in the present paper, it doesn't follow that economic phenomena *in fact* involve hypercomputation.[⋆] This can be seen immediately by reflecting for a moment upon the fundamental results from mathematical logic we alluded to above. For example, let h be the halting function from indices i for Turing machines[†] paired with natural numbers n given to such machines as input, to 1 or 0 as a verdict on whether or not Turing machine M_i halts or not after being started with n as input on its tape. As is well known, the *Entscheidungsproblem* is Turing-solvable if and only if h is Turing-computable. This shows immediately that it *might* be that hypercomputation is necessary in order to provide an accurate model of mathematical reasoning. This might be the case because human mathematicians (and others working in the formal sciences) might, for all we know, be capable of solving the *Entscheidungsproblem*. Of course, it could turn out that great discoveries in mathematics, for example Wiles' proof of Fermat's Last Theorem [48, 49], occur in part because the specific instances of the general problem just happen to be within human reach, while the arbitrary instance isn't.[‡]

The plan of the paper is entirely straightforward. We simply explain how some formal modeling of economic phenomena may well naturally invoke hypercomputation. After providing an overview of a "science-of-science" perspective from which we here view economics (§2), we touch upon modeling in the following three areas, viz.,

1. the Chain Store Paradox, the modeling of which can be naturally carried out in logics wherein proof-finding is beyond Turing machines (§3);

2. use of so-called "mental" logics to model real-world human reasoning and decision-making in ways more accurate than those enabled by standard, normatively correct logics (§4);

3. economic planning and prediction modeled using interacting Turing machines (§5); and

4. modeling certain macro-economic questions, specifically *economic collapse* (§6).

[⋆] In a parallel, while a formal model of physical reality may involve information-processing above the Turing Limit, it doesn't follow from this that physical reality is in fact hypercomputational in nature. We suppose this is why despite such super-Turing models as those presented in [32], many scientists refuse to accept the proposition that hypercomputation is physically real.

[†] Here and hereafter herein, when we refer to 'Turing machines' independent of specification we leave open which particular definition (e.g., quaduple vs. quintuple transition rules, starting and ending conventions, etc.) of such devices is used in the analysis. Accordingly, any and all generic assertions regarding TMs, if completely formalized, would of course have recourse to all the options for specifying the particulars of TMs.

[‡] In laying down at the outset an open-mindedness as to whether or not hypercomputation is in fact active in the phenomena modeled in economics we leave aside consideration of some of our own work. E.g., Bringsjord in [7, 8, 9] has argued that human persons hypercompute.

We leave for another day consideration of the question of whether such modeling can be non-question-beggingly dispensed with in favor of mere Turing-level modeling without leaving aside crucial phenomena. It is easy enough to simplify economic phenomena in such a way that they can be modeled using elementary computation, as long as one is willing to brutally simplify. For example, Cottrell et al. in [12] explain that "modernized" versions of socialist planning in the general style of Oskar Lange (e.g. see [26]) can be viewed as standard computation as long as we simply assume that human persons are themselves incapable of hypercomputation. But of course if human cognition itself involves hypercomputation, say in the creation of new products, it would hardly be possible for a Turing-level planning algorithm to model human innovation through time. (These issues are to some degree taken up later in section 5.) We also leave aside, due to space constraints, many additional economic phenomena the formal modeling of which might require hypercomputation.

2 THE SCIENCE OF SCIENCES

This section clarifies our overall (rather humble) goal for the paper.

The goal of the rational scientific enterprise can be thought of as that of finding *models* that fit some observed data, with the hope that these models eventually accurately describe the underlying reality from which these data flow. There has been significant work in formally studying and understanding the scientific process, a study we call *the formal science of science*.[¶] For example, an introduction to a particularly promising computational science of science can be found in Jain's [20], which Bringsjord and Sundar G. use in teaching the formal science of science.

One of the first scientific paradigms considered in [20] is *Language Identification* involving two entities, Nature and a scientist. [§] A language L is a set of finite strings composed from a finite alphabet Σ, that is $L \subseteq \Sigma^*$. A language represents some subset of the natural numbers and can be used to represent a posssible reality.[||] Only the recursively enumerable languages $\mathcal{E}$ are considered in [20] (which is of course already a presupposition against hypercomputation). A scientist $\mathbf{F}$ operates in some fixed and unknown reality L_t and gets data from nature in the form of strings from the language L_t. The job of the scientist is to produce hypotheses about the possible true language from which strings are presented to it. A scientist is modeled as any function $\mathbf{F} : SEQ \rightarrow \mathbf{N}$; where SEQ is the sequence of all finite strings that nature can produce. The output of the scientist is a natural number which corresponds to a representation of some partial recursive function in some *programming system* ν. A scientist encodes a language L using a natural number i by using the set of strings accepted by the ν program i, written as $W_i^\nu = L$.

Initially, Nature prepares an enumeration T of $L_t \cup \{\#\}$, where $\#$ is the blank symbol. Nature then presents the elements of T, also known as a *text* for L_t, sequentially to the scientist at each instant in a discretized timeline. The scientist then produces a hypothesis or a conjecture after examing the data it has seen from Nature so far. Figure 1 illustrates this process.

In our opinion, this framework conspicuously lacks a few aspects which are ubiquitous in science. One missing aspect is the concept of a proof (or at least a justification or explanation) for the conjectured hypothesis. This suggests the need for an improved formal science-of-science framework that handles declarative information and can produce proofs for conjectured hypotheses. Figure 2 contains an overview of one such possible improved framework. As should be clear from the figure, the basic new idea is that our scientist must justify its conjectures. This happens because the scientist attempts to ascertain whether or not some conjecture χ follows from a formal theory Φ, as confirmed by some proof $\mathcal{P}$. If the answer is in the affirmative, χ is added to the knowledge-base for the science in question.

2.1 Science of Sciences applied to Economics

In accordance with our motivation for seeking a plausible framework for a science of science, and specifically an instance of such a framework to model the science of economics, we outline version 1 of Bringsjord's science-of-economics framework $\mathbf{F}_{econ}$ (see Figure 3). In this framework, the scientist $\mathbf{F}_{econ}$ produces a machine $\mathcal{M}$ (which

[¶] Please note that we say *formal*. In e.g. the philosophy of science there has long been somewhat systematic consideration of what science is, how discovery within it works, and so on. For a non-technical yet crisp introduction to philosophy of science from a formalist see [11].

[§] The notion of equating science with language identification is analogous to equating computation with language recognition.

[||] Here we assume that physical measurements can be represented with arbitrary precision using the rational numbers, which in turn can be represented using the natural numbers.

FIGURE 1
Conventional CLT-Based Science of Science

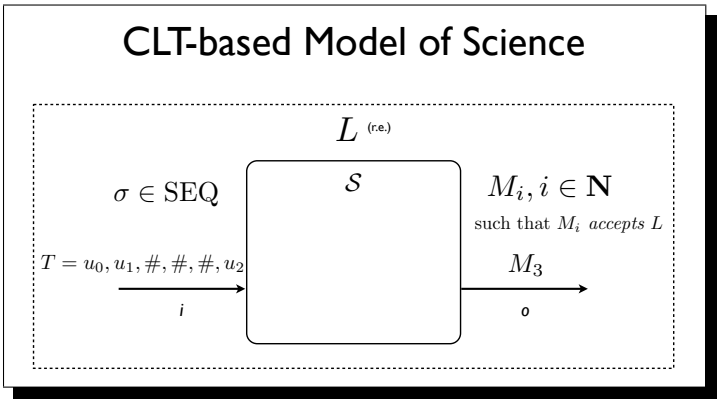
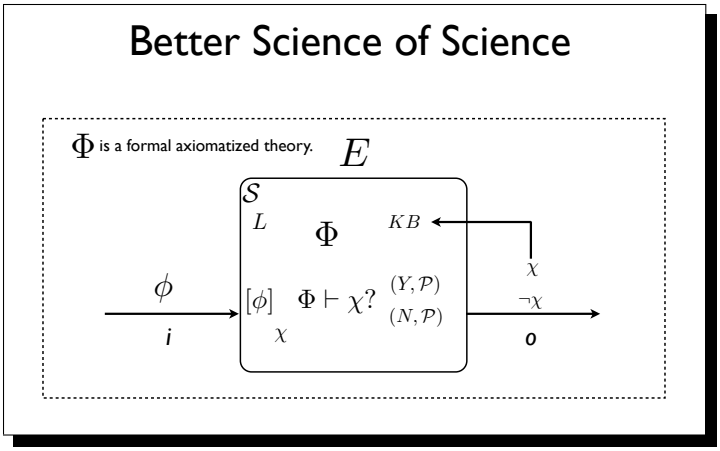


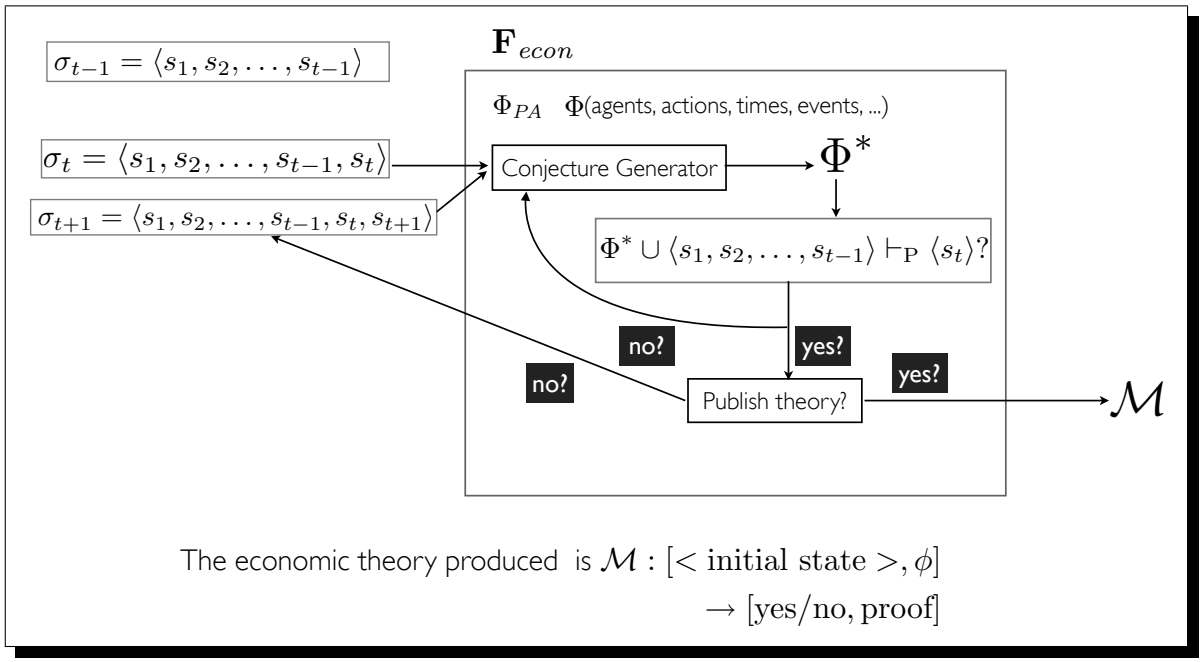
FIGURE 2
Improved Computational Learning Theory



need not be a Turing machine) that is capable of taking in a set of initial or prior states of the relevant market, and answers whether a certain future state-of-affairs ϕ will obtain or not in the future (or whether such a state-of-affairs is possible in the future, etc.), and a proof that supports its answer. The processing that allows $\mathcal{M}$ to be produced includes consideration of sequences of states s_i of relevant markets. (We write $\langle s_i \rangle$ to indicate that a state has been expressed in declarative form in the form of as set of formulas.) Note that in the general case involving only inexpressive extensional logics like first-order logic this problem is Turing-unsolvable. Hence we don't impose the restriction that $\mathcal{M}$ be a Turing machine. To emit $\mathcal{M}$, the scientist takes in a series of declarative statements σ_t about the past states of the world and uses a conjecture generator to conjecture a theory Φ^* which it tests on the available data. If the conjecture agrees with the available data, the scientist then decides either to publish the theory in the form of $\mathcal{M}$ or examine more data. If the conjecture does not agree with the available data, the scientist then revisits the conjecture-generation stage. We assume that Φ^* is expressive enough to account for basic arithmetic, agents, agent actions, agent goals, agent percepts, times, events etc. Again, note that in the general case both $\mathbf{F}_{econ}$ and $\mathcal{M}$ can be Turing-unsolvable.

In the specific context of economics, we seek to produce models of $\mathbf{F}_{econ}$ and $\mathcal{M}$ for modeling specific economic phenomena, but which might require hypercomputational processes. This *does not* imply that economic phenomena observed so far *are* hypercomputational. Once again, our local objective is to show the *possibility* of

FIGURE 3
Science of Sciences applied to Economics



hypercomputation in economic phenomena.

3 THE CHAIN STORE PARADOX

Reinhard Selten (see [38]), Nobel laureate in economics, introduced the Chain Store Paradox (CSP); it provides an excellent example of phenomena the fastidious formal modeling of which may well require hypercomputation.

CSP centers around strategic interaction between a "chain store" (e.g., Wal-Mart, McDonald's, or even, say, Microsoft) and those who may attempt to enter the relevant market and compete against this store. The game here is an n-stage, $n+1$ one, in which the $n+1^{th}$ player is our chain store (CS), and the remaining players are the potential entrants $E_1, E_2, \ldots, E_n$. At the beginning of the k^{th} stage, E_k observes the outcome of the prior $k-1$ stages, and chooses between two actions: `Stay Out` or `Enter`. An entrant E_k opting for `Stay Out` receives a payoff of c, while CS receives a payoff of a. If, on the other hand, E_k decides to enter the market, CS has two options: `Fight` or `Acquiesce`. In the case where CS fights, CS and E_k receive a benefit of d; when CS acquiesces, both receive b. Values are constrained by $a > b > c > d$, and in fact we shall here set the values, without loss of generality, to be, respectively, 5, 2, 1, and 0. Please see Figures 4 and 5, which provide snapshots of early stages in the game. We specifically draw your attention to something that will be exploited later in this section, which is shown in Figure 4: viz., that we have used diagonal dotted lines, with labels, to indicate key timepoints in the action.

But why is CSP called a *paradox*? Please note that there are at least three senses of 'paradox' used in formal logic and in the formal sciences generally, historically speaking. In the first sense, a paradox consists in the fact that it's possible to deduce some contradiction $\phi \wedge \neg\phi$ from what at least seems to be a true set of axioms or premises. A famous example in this category of paradox is Russell's Paradox, which pivots on the fact that in standard first-order logic

$$\exists x \forall y (Rxy \leftrightarrow \neg Ryy) \vdash \phi \wedge \neg\phi.$$

As Frege learned, much to his chagrin, Russell's Paradox demonstrated that Frege's proposed meta-mathematical

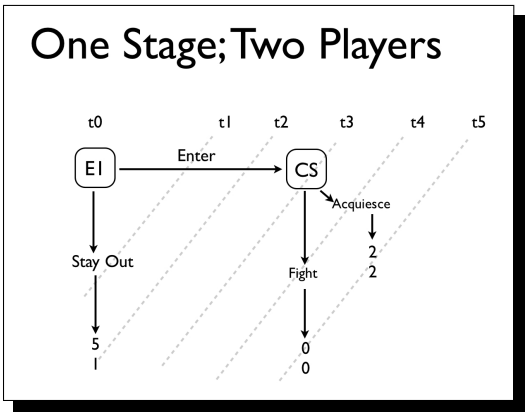


FIGURE 4
One-Stage, Two-Player Snapshot in CSP

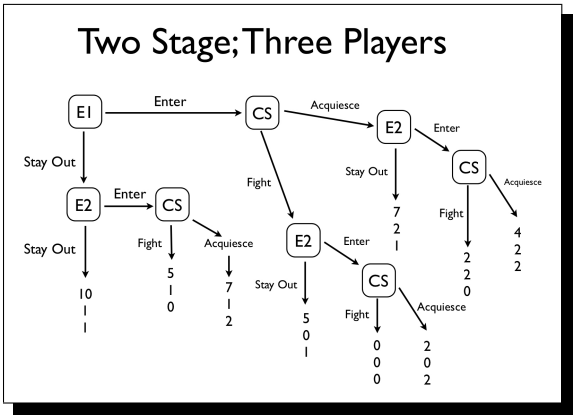


FIGURE 5
Two-Stage, Three-Player Snapshot in CSP

set-theoretic foundation for mathematics was inconsistent.[#] (Other examples of paradoxes of this form that are relevant to economics are the Lottery Paradox and the Paradox of the Preface, but discussing these would take us too far afield. Both of these paradoxes are in our opinion solved by Pollock in [31], using his Oscar system.)

In the second sense of 'paradox' in logic, a theorem is simply regarded by many to be extremely counter-intuitive, but no outright contradiction is involved. A famous example in this category is Skolem's Paradox, elegantly discussed in [15]. Finally, in the third sense of 'paradox,' a contradiction is produced, but not by a derivation from a single body of unified knowledge; rather, the contradiction is produced by deduction of ϕ from one body of declarative knowledge (or axiom set, if things are fully formal), and by deduction of $\neg\phi$ from *another* body of declarative knowledge (or outright axiom set, in the fully formal version, if there is one). Additionally, both bodies of knowledge are independently plausible, to a very high degree. A famous example of a paradox in this third sense — quite relevant to decision theory and economics, but for sheer space-conservation reasons outside of our present discussion — is Newcomb's Paradox; see for example the first published treatment, [29]. It is into this third category of 'paradox' that CSP falls.

More specifically, we have first the following definition and theorem. (Selten himself doesn't provide a fully

[#] A classic presentation of axiomatic set theory as emerging from the situation in which naïve set theory was plagued by Russell's Paradox (and others of the first type) is [43]. Nice discussion of Russell's Paradox can be found in [4].

explicit proof of the theorem in question. For more formal treatments, and proofs of backward induction, see e.g.,[2, 3]. In the interest of economy, we provide only a proof sketch here, and likewise for the theorem thereafter for deterrence.)

Definition (GT-Rationality): We say that an agent is **GT-rational** if it knows all the axioms of standard game theory, and all its actions abide by these axioms.

Theorem (GT-Rationality Implies Enter & Acquiesce): In a chain-store game, a GT-rational entrant E_k will always opt for `Enter`, and a GT-rational chain store CS will always opt for `Acquiesce` in response.

Proof-Sketch: Selten's original strategy was "backward induction," which essentially runs as follows when starting with the "endpoint" of 20 as he did. Set $k = 20$. If E_{20} chooses `Enter` and CS `Fight`, then CS receives 0. If, on the other hand, CS chooses `Acquiesce`, CS gets 2. Ergo, by GT-rationality, CS must choose `Acquiesce`. Given the common-knowledge supposition in the theorem, E_{20} knows that CS is rational and will acquiesce. Hence E_{20} enters because he receives 2 (rather than 1). But now E_{19} will know the reasoning and analysis just given from E_{20}'s perspective, and so will as a GT-rational agent opt herself for `Enter`. But then parallel reasoning works for $E_{18}, E_{17}, \ldots, E_1$. **QED**

The other side of the "third-sense" paradox in the case of CSP begins to be visible when one ascertains what real people in the real world would do when they themselves are in the position of CS in the chain-store game. As has been noted by many in business, such people are actually inclined to fight those who seek to enter — and looking at real-world corporate behavior shows that fighting, at least for some initial period of time, is the strategy most often selected.** From our formal science-of-science perspective, and specifically from the perspective of the CLT-based science of sciences framework shown in Figure 3, these empirical factors are of high relevance because they are to be predicted by the machine $\mathcal{M}$ given as output by $\mathbf{F}_{econ}$. Indeed, the very purpose of $\mathcal{M}$ is to predict future states of the world on the strength of the declarative representation of past states and/or present state, along with declarative information about agents, and their goals, beliefs, perceptions, possible actions, and so on.

Sure enough, there does appear to be a formal rationale in favor of thinking that such a prediction machine as $\mathcal{M}$ would predict fighting. This rationale is bound up inseparably with *deterrence*, and can be expressed by what can be called *forward induction*. The basic idea is perfectly straightforward, and can be expressed via the following definition and theorem (which for lack of space we keep, like its predecessor, somewhat informal and compressed).

Definition (Perception; m-Learning; Two-Option Rationality): We say that an agent α_1 is **perceptive** if, whenever an agent α_2 performs some action, α_1 knows that α_2 does so. We say that an agent α is an **m-learner** provided that when it sees agents perform some action `A` m times, in each case in exactly the same circumstances, it will believe that all agents into the future will perform `A` in these circumstances. And we say that an agent is **two-option-rational** if and only if when faced exclusively with two mutually exclusive options `A1` and `A2`, where the payoff for the first is greater than the second, that agent will select `A1`.

Theorem (Rationality of Deterrence from CS): Suppose we have a chain-store game based on n perceptive agents, each of whom are m-learners. Then after m stages of a chain-store game in which each potential entrant seeks to enter and CS fights, the game will continue indefinitely under the pattern of `Stay Out`.

Proof-Sketch: The argument is by induction on **N** (natural numbers). Suppose that the antecedents of the theorem hold. Then at stage $m + 1$ all future potential entrants will believe that by seeking to enter, their payoff will be 0, since they will believe that CS will invariably fight in the future in response to an enter-

** Selten in [38] prophetically remarked that he never encountered someone who said "he would behave according to induction theory [were he the Chain Store]."

ing agent. In addition, as these agents are all two-option-rational, they will forever choose `Stay Out`. **QED**

We see here that the two previous theorems, together, constitute a paradox in the aforementioned third sense.[††] In general, paradoxes of the third type can be solved if one simply affirms one of the two bodies of knowledge, and rejects the other. However, we are under no requirement to take a stand. In fact, the key fact for our purposes in the present paper is that no matter which body of knowledge one prefers over the other, the process of doing economics (i.e., what our $\mathbf{F}_{econ}$ does inside the box of Figure 3), and the artifacts produced by doing economics (i.e., by $\mathbf{F}_{econ}$), may well involve hypercomputation. This is easy to see, as follows.

One can fully formalize and prove a range of both "highly-expressive" backward induction and deterrence theorems under the relevant assumptions. These theorems are differentiated by way of the level of expressivity of the underlying logics used. The calculi use more detailed calculi than have been used before in the chain-store literature in mathematical economics. For example, it is possible to prove a version of both the induction and deterrence theorems using an "economic cognitive event calculus" ($\mathcal{ECEC}$) based on the cognitive event calculus in [1], which allows for epistemic operators to apply to sub-formulas in full-blown quantified modal logic, in which the event calculus is encoded.

Any agent (e.g., $\mathbf{F}_{econ}$) able to determine in the arbitrary case whether or not some proposition holds in $\mathcal{ECEC}$ under some axiom set would be capable of hypercomputation, and would therefore, if accurately modeled, require hypercomputation. From the standpoint of theoretical computer science and logic, all versions of the chain-store game, hitherto, have involved exceedingly simple formalizations of what agents know and believe in this game, and of change through time. To see this more specifically, note that in standard axioms used in game theory, for example in standard textbooks like [30], knowledge is interpreted as a simple function, rather than as an operator that can range over very expressive formulas. This same limited, simple treatment of knowledge and belief is the standard fare in economics. Yet, it can be shown that the difficulty of computing, from the standpoint of some potential entrant E_k or chain store CS in some version of a chain-store game, is at the level, minimally, of Σ_1 when attempting to compute whether by induction or deterrence they should opt for `Enter` or `Stay Out`.

We conclude this section by directing your attention to Figure 6. This figure shows a proof in $\mathcal{ECEC}$, implemented and machine-checked in the Slate system [10], for the prediction that, from the start of the chain-store game, at the seventh timepoint (t_7), the third agent will `Stay Out`. Note that the production of this proof constitutes a simulation that demonstrates that this future timepoint will be reached from the initial state, and so the process here coincides nicely with the process summarized pictorially in Figure 3. General predictions of this kind, made by the imagined $\mathcal{M}$ produced by our idealized economist $\mathbf{F}_{econ}$, would obviously require hypercomputation, for the simple reason that $\mathcal{ECEC}$ inherits the undecidability of first-order logic.

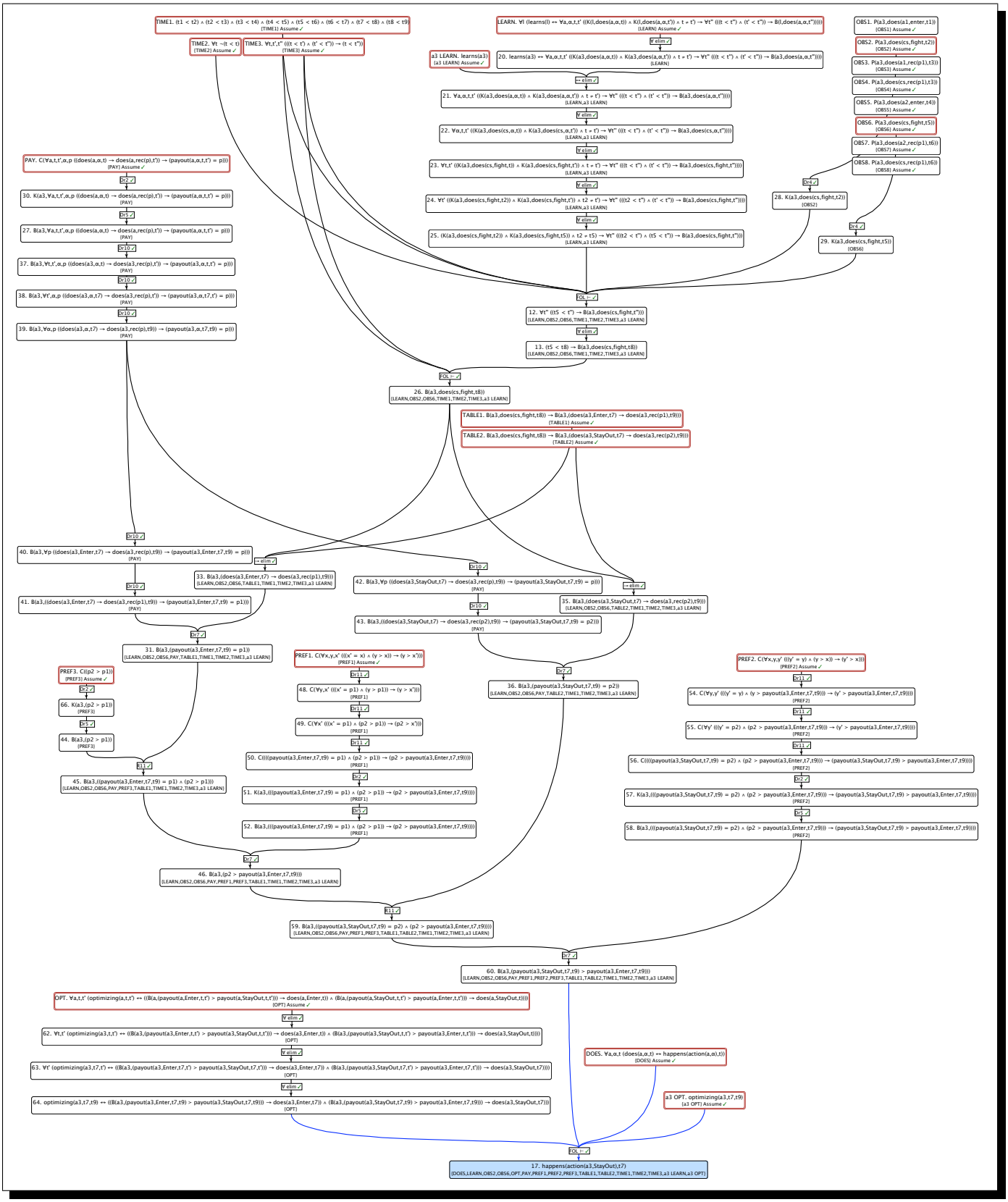
4 MENTAL DECISION LOGIC

Jones, suppose, is trying to decide which of two cars to purchase. One car is a so-called sport-utility vehicle (SUV), and the other option is a sedan. In light of the key possibility that his drives will often take him through the snowy (Berkshire and Taconic) mountains, which vehicle ought he to buy?

We can represent such a decision problem using the machinery of some established decision theory. The theory we prefer happens to be a merging of the decision theory of Savage (e.g. see [37]), often embraced by economists, with the decision theory of Jeffrey [21], often welcomed by those in related fields (logic and computer science, e.g.) inclined toward logic-based modeling. The hybrid theory is Joyce's, given in [23]. However, in the interests of space, we severely compress the language of Joyce, and regard a decision problem $\mathcal{D}$ as only the problem of deciding which one of two competing actions a_1 and a_2 provides more utility to an agent s in the light of a set Φ of logical formulas. (When actions appear as arguments to $>$ we assume that it's the utility of the actions that are being related.) Φ is assumed to include information about the world, about causal connections between actions

[††] We would be remiss if we didn't mention two points of scholarship, to wit: (1) Game theorists have long proposed modifications or elaborations of the chain-store game which allow game theory to reflect sensitivity to the cogency of the deterrence line of reasoning. E.g., see [24]. (2) Innovative approaches to CSP that in some sense opt for a third approach separate from both backward induction and deterrence have been proposed. E.g., see the one based in evolutionary computation proposed by Tracy in his [46].

FIGURE 6
An entrant's action in the Slate system. We are grateful to Joshua Taylor, Chief Developer of Slate, for helping with this proof and implementing it in Slate.



and outcomes, about the beliefs of s, and so on; recall the comprehensive calculus Φ^* postulated in Figure 3. We thus write a decision problem $\mathcal{D}$ as a quadruple (Φ, s, a_1, a_2), and such a problem would for example be solved if it could be determined for or by s that $\Phi \vdash^C a_1 > a_2$.

But what exactly does it mean to say that $\Phi \vdash^C a_1 > a_2$? In general, this means that $a_1 > a_2$ can be proved from Φ. However, we can only genuinely answer this question if we have defined the context; if, that is, we have defined the background logic and proof calculus C. Suppose for example that the background logic is standard first-order logic and that C is some standard proof calculus for first-order logic — say $\mathcal{F}$ from [4], or the proof calculus of our own Slate system (. [10]) Then the meaning of a decision problem is clear. But it's now well-known in cognitive science that if this background logic and associated calculus is a normatively correct one (such as first-order logic), the modeling will fail, for the simple reason that even the vast majority of college-educated adults are incapable of reasoning in normatively correct deductive fashion. (For a nice survey, see [42].) This brute fact, quite in line with what is today called "behavioral" economics, has catalyzed the creation of "mental" logics: logics that are specifically tailored to reflect how human beings (save for e.g. logicians and so on) reason. Researchers who have invented and refined such logics include Braine in [6], Johnson-Laird in [22], Rips in [34], and Yang et al. in [50]. Of these, Yang (in e.g. [51]) has recently presented a logic (*mental decision logic*; 'MDL' for short) designed specifically for representing decision-making in scenarios commonly studied by economists.

Let's refer to the space of mental logics as $\mathcal{ML}$; and let $ML \in \mathcal{ML}$ be without loss of generality here be some mental logic whose inferential machinery is C', and let Φ_{ML} be some set of formulas expressed in the language of ML which captures the above situation involving Jones, where $a_1 > a_2$ is one such formula. (Note that we must speak of "inferential machinery" rather than a standard proof calculus (or theory). This is the case because what can be inferred in C' departs from standard deductive inference.) At present, so far as we know, it's an open question as to whether $\Phi_{ML} \vdash^{C'} a_1 > a_2$ in the general case is decidable using an ordinary Turing machine.[‡‡] Given this, and given that accurate predictions about Jones's behavior by scientist $\mathbf{F}_{econ}$ may require a capacity to judge whether $a_1 > a_2$ can be inferred from the knowledge base in question, it may be that hypercomputation on the decision-theoretic front is part of rigorous economics.

5 ECONOMIC PLANNING AND PREDICTION VIA TURING-LEVEL ACTORS

Minimally, an economic system can be modeled as consisting of a planning agent which tries to control the economy so that some utility is maximized, a set of independent economic agents, and an environment which provides a setting in which these agents function. In this particular scenario, $\mathbf{F}_{econ}$ plays not only the role of a scientist but also that of a *planner* in an economy. $\mathbf{F}_{econ}$ acts in the economy it studies to increase some utility or reward. The purpose of this section is to argue for the following claim.

***Claim**: Even if real economies and agents in those economies can be modeled as Turing machines, formal solutions to non-trivial planning problems can be Turing unsolvable.*

To model such agents, we look at Hutter's computational model: the *AIXI Model* [18] for an agent seeking to maximize the reward that it obtains in an unknown environment. The purpose of Hutter's model is to provide a formal model for artificial intelligence (AI). This model can be adapted with some extensions to specifically model an economic planner seeking to derive maximum utility in an economy comprised of zero or more other cognitive agents. Hutter's model is composed of simple conceptual ingredients and can be naturally adapted to the economic domain. We start with a condensed, yet informal, description of Hutter's model, closely following notation used in [18]. A simpler description can be found in [27]. Detailed descriptions, definitions and proofs can be found in [17, 18, 19].

[‡‡] That this is an open question may not be due to the technical difficulty of obtaining proofs that settle the question for some $\mathcal{L}$ and C. Openness may be due to other factors, e.g., to the fact that mental logics are often left too imprecise for the decidability question to be posed with sufficient rigor.

5.1 Description of Hutter's AIXI model

Hutter's AIXI model is a formal model of an agent operating in an unknown environment. Given certain reasonable assumptions, this model is universally optimal across all possible environments and has certain provable optimality properties. This model builds upon Solomonoff's Theory of Induction [41] and Sequential Decision theory [44]. Before we describe Hutter's AIXI model, we present some necessary preliminaries.

Preliminaries

We are concerned with sequence-prediction tasks in the AIXI model. Any prediction task can be modeled as a sequence prediction task. In the AIXI model, the agent will observe the sequence $x_1, \ldots, x_{t-1}$ from the enviroment, which corresponds to percepts that an agent will obtain in an enviroment. The agent then predicts x_t and then outputs an action y_t^* which maximizes its expected reward. The following concepts play a role in the specification of the optimal action y_t^*. Following Hutter, we denote the sequence $x_1, x_2, \ldots, x_n$ by $x_{1:n}$ and the sequence $x_1, x_2, \ldots, x_{n-1}$ by $x_{<n}$

Bayesian Mixture Distribution We have some data x which is explained/produced by a hypothesis/model $H^* \in \mathcal{M}$. We don't know the true model producing our data. We have prior probability for model H_i given by $prior(H_i)$. The Bayesian mixture distribution of H_i with respect to the prior probabilities $prior(H_i)$ is specified by

$$\sum_i prior(H_i) P(x|H_i)$$

Prefix Code A prefix code P is a set of strings such that no string in P is the prefix of another string in P.

Monotone Universal Turing Machine A monotone Turing machine is a Turing machine with *1*) one or more bi-directional working tapes; *2*) a unidirectional input tape which is read-only and left-to-right; and *3*) a unidirectional output tape which is write-only and left-to-right. The unidirectional input and output tapes are necessary to make the technical task of imposing chronological conditions on Turing machines easier.

Minimal Programs When the input tape of a monotone universal Turing machine U contains p to the left of the input head when the last bit of a string x is output, we write $U(p) = x*$. The set of all such strings p forms a prefix code, and the codes are called *minimal programs* for x.

Universal Prior The *universal prior* is the probability that the output of a universal monotone Turing machine U starts with the string x when provided a random input drawn uniformly from the space of all inputs. We consider all possible minimal programs p for which the Turing machine U has output x. The universal prior M is defined as

$$M(x) = \sum_{p:U(p)=x*} 2^{-l(p)} \tag{1}$$

Intuitively, the above equation can be interpreted as a Bayesian mixture distribution. The model class is the set of all programs p which produce $x*$. If we consider deterministic programs, the probability of a minimal program for x producing x is 1. In accordance with Occam's razor, the prior for any particular model is $2^{-l(p)}$.

Convergence of the Universal Prior If the true distribution μ from which the sequence is drawn is a Turing computable distribution, the posterior universal prior $M(x_t|x_{<t})$ converges to the true posterior distribution $\mu(x_t|x_{<t})$.

Generalized Universal Prior If the model class $\mathcal{M}$ is the class of all enumerable semi-measures then the Bayesian mixture obtained by using the prior $2^{-K(\nu)}$ for $\nu \in \mathcal{M}$ is denoted by ξ and is defined as

$$\xi(x) = \sum_{\nu \in \mathcal{M}} 2^{-K(\nu)} \nu(x) \tag{2}$$

$K(\nu)$ is the Kolmogorov complexity of ν, and this factor penalizes more complex distributions. An enumerable function is one for which lower bounds can be finitely computable, but the function itself need not

be finitely computable. A measure is a semi-measure which is also normalized. The generalized prior is a Bayesian mixture of enumerable (not just finitely computable) semi-measures (not just measures) and takes into account more distributions than the universal prior defined above. The generalized universal prior ξ is also an enumerable semi-measure.

Convergence of ξ It can be show that the posterior distribution $\xi(x_t|x_{<t})$ converges to the true posterior distribution $\mu(x_t|x_{<t})$ in a finite amount of time if μ is enumerable.

The AIXI Model

The AIXI model is an agent-based model in which an agent P performs an action $y_t \in \mathcal{Y}$ in an environment Q at time t. The environment responds with $x_t \in \mathcal{X}$, which can be split into a unique percept $o_t = o(x_t)$ and a reward $r_t = r(x_t)$. This continues till some time m termed as the *horizon* or the *lifetime* of the agent.

We assume that the environment is modeled by a true probability distribution μ which is also enumerable. The probability of the environment producing the percept sequence $x_{1:n}$ is $\mu(x_{1:n})$, which can be reasonably approximated by $\xi(x_{1:n})$ if μ is computable. Now consider the agent at time k. The expected reward at time $k+1$ if the agent chooses action $\overline{y}$ is

$$R(k+1) = \sum_{x \in \mathcal{X}} r(x)\mu(x_{k+1} = x|x_{1:k}, y_{<k}\overline{y}) \tag{3}$$

The action to maximize the expected reward at time $k+1$ (denoted by y^*_{k+1}) is then given by the action which maximizes the above equation:

$$y^*_{k+1} = \arg\max_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} r(x)\mu(x_{k+1} = x|x_{1:k}, y_{<k}y) \tag{4}$$

A rational agent will seek to maximize the rewards that it expects over its entire lifetime and not just the reward it expects in the next time step. The expected reward to be obtained in cycles $k+1$ to m, given that agent performs actions $\overline{y}$ at time $k+1$, is given by this recursive equation:

$$R(k+1:m) = \sum_{x \in \mathcal{X}} [r(x) + R(k+2:m)]\mu(x_{k+1} = x|x_{1:k}, y_{<k}\overline{y}) \tag{5}$$

The action to maximize the expected reward to be obtained in cycles $k+1$ to m is then

$$y^*_{k+1} = \arg\max_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} [r(x) + R(k+2:m)]\mu(x_{k+1} = x|x_{1:k}, y_{<k}y) \tag{6}$$

The AIXI model is then obtained by replacing the unknown μ by ξ in the above equation. The AIXI Model is an ExpectiMax algorithm [36]. Unfolding the recursive equation and renaming the corresponding indices we get the following equation which specifies the best action to choose at time k.

$$y^*_k = \arg\max_{y_k} \sum_{x_k} \ldots \max_{y_m} \sum_{x_m} (r_k + \cdots + r_m).\xi(x_{k:m}|x_{<k}, y_{<k}y)$$

Properties of AIXI

AIXI is Turing-uncomputable. Intuitively, this is due to the involvement of the Turing-uncomputable Kolmogorov complexity and an infinite sum in the specification of AIXI's optimal action. The AIXI model has many associated optimality properties, and Hutter calls the AIXI model *universally optimal* as it can perform better than any other agent and can perform well in any environment.[¶¶] For proofs of these properties please consult [18].

5.2 Generalized Economic Planning is an Enriched AIXI Problem

Hutter's model deals with a rational agent seeking to maximize the rewards it obtains in an unknown environment. It is easy to generalize the AIXI model to that of a rational agent seeking to maximize some reward function in an environment containing some other cognitive agents. The agent which seeks to control the economy and

[¶¶] With some caveats.

maximize the reward/utility function models a planner in an economy and the other agents model participants in the economy. We model the planner, constituent agents in the economy, and the environment/economy as monotone Turing machines.

Our model, dubbed the *Economic-AIXI* model, consists of

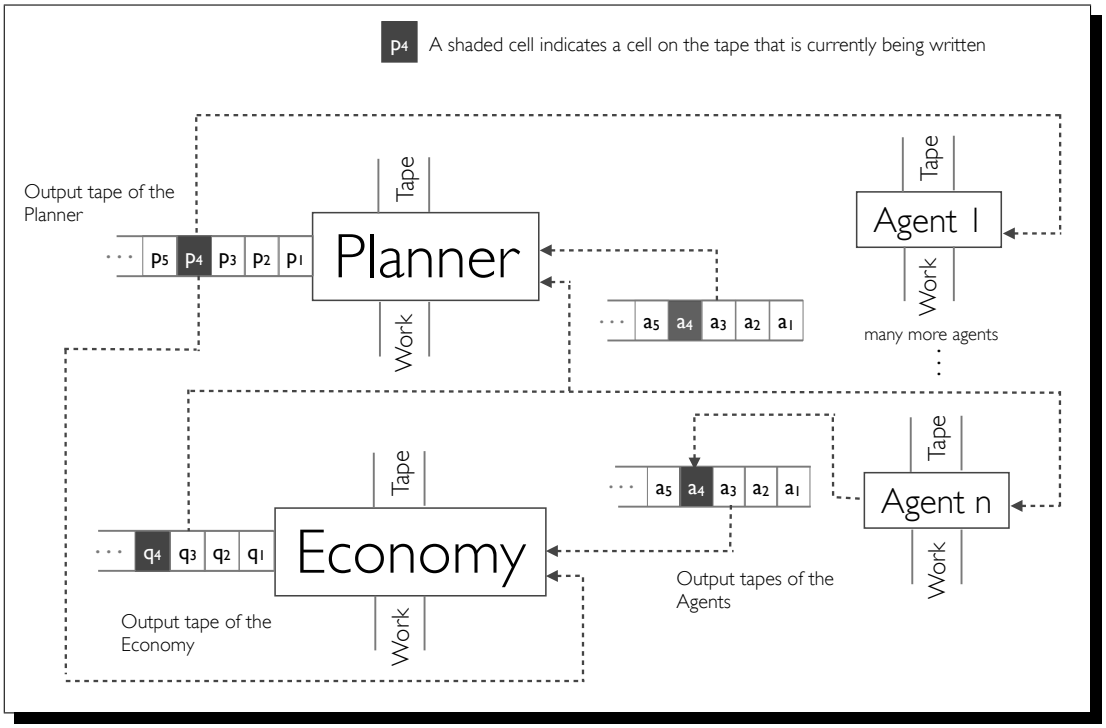
1. A central planner in the economy. The planner outputs plans to control the economy and the goal of the planner is to maximize the reward that it obtains from the economy.

2. A set of agents which constitute the economy $\mathcal{A} = \{A^1, \ldots, A^n\}$. The agents can be either producers or consumers in the economy, or both. The agents in the economy can either ignore or adhere partially or fully to the planner's commands.

3. The overall economy. The economy models interactions of different agents, natural resources, and consumption and production in the economy. The economy represents inanimate forces that act in an economy, such as laws of nature, forces outside the economy, etc.

Figure 7 shows a schematic description of our model. At time k, the planner reads symbols $x^1_{k-1}, \ldots, x^n_{k-1}$ from the output tapes of the agents $1 \ldots n$ and the symbol e_{k-1} from the output tape of the economy. The planner then writes p_k on its output tape. Agent i, where i ranges from $[1, \ldots, n]$, reads e_{k-1} from the output tape of the economy and p_k from the output tape of the planner and then writes the symbol x^i_k on its output tape. The economy reads symbols $x^1_{k-1}, \ldots, x^n_{k-1}$ from the output tapes of the agents and the symbol p_k from the output tape of the planner and then writes e_k on its output tape.

FIGURE 7
Economic-AIXI: Economic Planning Formalized via Primitives in Theory of Computation



At time instance t, the planner outputs plan p_t based on prior actions $(x^1_{<t}, \ldots, x^n_{t<})$ by different agents in the economy and the past states of the economy characterized by $e_{<t}$ on the output tape of the economy. The agents compute their next action based on the plan history $p_{1:t}$ and the economic history $e_{<t}$. The economy then moves

TABLE 1
Economic-AIXI entities and their specification

Economic Entity	Action
Agent i	$A^i(p_{1:t}, e_{<t}) = x^i_{1:t}$.
Economy	$Q(x^1_{<t}, \ldots, x^n_{<t}, p_{1:t}) = e_t$
Planner	$P(x^1_{<t}, \ldots, x^n_{<t}, e_{<t}) = p_{1:t}$

to a different state based on the past actions of the agents $(x^1_{<t}, \ldots, x^n_{t<})$ and the planner's actions $p_{1:n}$. Table 5.2 is a specification of each economic entity and the input-output behavior of the function it computes.

Our model has very general assumptions, viz., the entities in an economy are Turing-computable functions. With this minimal assumption we can model almost any economic problem. For example, consider an agent which seeks to maximize capital gains by trading stock in some market. The economy for this agent is the set of companies trading in that market and factors which affect the fortunes of those companies. The agents $\mathcal{A}$ are other traders in the market. This same model can be applied to a planner of a national economy seeking to increase the GDP of his nation. The agents can then be used to model different centers of production and the economy can be used to specify the relationship between the consumption and production of different products.

The Turing-unsolvability of the Economic-AIXI model follows from the Turing-unsolvability of the AIXI model.

6 MODELING ECONOMIC COLLAPSE

At present, even though there are no formal models of economic collapse, there are informative, informal notions of what an economic collapse might look like (for relevant material see [5, 28, 25, 14, 35]). Wikipedia, for example, has the following synoptic yet informative account of what an economic collapse looks like.

> An economic collapse is a devastating breakdown of a national, regional, or territorial economy. It is essentially a severe economic depression characterized by a sharp increase in bankruptcy and unemployment. A full or near-full economic collapse is often quickly followed by months, years, or even decades of economic depression, social chaos and civil unrest. Such crises have both been seen to afflict capitalist market economies and state controlled economies. [47]

Obviously, we should be highly motivated to prevent economies from collapsing, because the associated negative affects will harshly impact many lives. Currently, at least to our knowledge, commonly recognized formal theories of economic collapse do not exist. This is so, among other reasons, because economies are enormously complex and there can be many causes for their collapse, and because much of economics remains highly informal (at least relative to the success stories that inspire us; e.g., key results in mathematical and philosophical logic). We will present and analyze a very abstract but powerful formal model of economies based on computability theory. Specifically, we note that economies (like any other discrete finite systems) can be modeled as Turing machines, which are believed to be one of the most powerful (if not the most powerful; it depends of course on whether one accepts hypercomputation or not) type of formal, computational system.

We now consider a scientist $\mathbf{F}_{econ}$ that seeks to model economic-collapse-related problems by producing Turing machines $\mathcal{M}$ which seek to (but need not) answer the following questions.

Economic Collapse Problem or ϕ_{ECP}: Will the economy E started with initial conditions $\mathcal{I}$ collapse at time t?

Generalized Economic Collapse Problem or ϕ_{GECP}: Will the economy E started with initial conditions $\mathcal{I}$ and after experiencing a history h collapse at some time t after history h?

Economic Immortality Problem or ϕ_{ECP}: Will the economy E started with initial conditions I ever collapse?

We show that in the general case a simulator $\mathcal{M}$ capable of answering the above questions with a proof, and a scientist F_{econ} that produces $\mathcal{M}$, would both have to be hypercomputational.

We assume that F_{econ} examines an economic system E and produces a Turing machine $\mathcal{M}$ that tries to answer one of the questions listed above. $\mathcal{M}$ has the following characteristics.

1. At a very abstract level, the maximum number of independent entities possible in an economy — such as the maximum number of companies that can be registered in the stock market, the maximum number of producers and consumers that are possible, the volume of transactions, existing regulations, interest rate, etc. — determine *in principle* the minimum number of states n of the Turing machine $\mathcal{M}$ that seeks to model that economy.

2. If the economy survives, its modeling TM $\mathcal{M}$ writes 1 in the current cell pointed to by the read/write head of the machine and moves one position to the left or to the right; otherwise the machine halts. For simplicity, let's assume that transitions from any state to any other state are possible (in reality, for the specific economic model, some transitions could be disallowed). Let's further assume that the set of final/halting states of the TM represents states of economic collapse; that is, after reaching the final state, the economy halts: no further moves are possible.

We will consider here only "hard" collapses; that is, no further moves in the economy are possible. It is possible to extend our model to allow "soft" collapses too, by adding the estimates of how far the TM is from the collapse/terminal state, and then avoiding approaching collapse states, or the recovery transitions from the collapse states to non-collapse states; but for simplicity we consider here "hard" collapses only.

6.1 Economies that Collapse

DEFINITION 6.1 (THE ECONOMY COLLAPSE PROBLEM (ECP)) *Let $ECP(n)$ be the maximum amount of time for which any economy with n states can function without collapsing. Here collapse can be equated with the corresponding Turing machine M halting when started on a blank tape. Compute $ECP(n)$ for arbitrary values of n.*

Usually when we study economic systems, we never start from time zero: we usually have some knowledge of the history of that particular system. (Note that $\mathbf{F}_{econ}$ explicitly factors in histories.) Accordingly:

DEFINITION 6.2 (THE GENERALIZED ECONOMY COLLAPSE PROBLEM ($GECP$)) *Let $GECP(n)$ be the maximum amount of time for which any economy with at most n states can function without collapsing, assuming that it functioned already for some finite amount of time. Compute $GECP(n)$ for arbitrary values of n.*

Readers familiar with computability theory can immediately recognize that the above problems are analogous to the busy-beaver function $\Sigma(n)$, max-shift function $S(n)$, and non-empty-tape-busy-beaver function $\Sigma(n, x)$, respectively [33]. These Turing-uncomputable functions can be reduced to each of the above problems and their Turing-uncomputability follows from that reduction.

To prove the unsolvability of ECP we recall a related problem [33]:

DEFINITION 6.3 (THE BUSY BEAVER PROBLEM (BBP)) *Consider a deterministic 1-tape Turing machine with the unary alphabet $\{1\}$ and the tape alphabet $\{1, B\}$, where B represents the tape's blank symbol. This Turing machine starts with an initial empty tape and accepts by halting. For an arbitrary (and, of course, finite) number of states $n = 0, 1, 2, ...$, find two functions: the maximum number of 1s written on the tape before halting (known as the busy-beaver function $\Sigma(n)$) and the maximum number of steps before halting (known as the maximum-shift function $S(n)$.*

Note that $\Sigma(n)$ represents the space complexity and $S(n)$ the time complexity of a BBP TM. Obviously, $\Sigma(n) \leq S(n)$. Both functions grow faster than any computable function; that is, the Busy Beaver Problem has been proven to be TM-unsolvable. If it were possible to solve the BBP, we could reduce the BBP to the halting problem of

a universal Turing machine (UTM), and we could then disprove the latter's undecidability. We could do this by computing the max-shift function $S(n)$ and running a UTM up to $S(n)$ steps. If the BBP TM halted during first $S(n)$ steps, then UTM would halt; otherwise it would run forever and the answer for UTM halting would be NO. In both cases, we would be able to decide in a finite number of steps: YES or NO regarding the halting of UTM. Of course, the algorithm would not be too practical because both functions grow very fast and their precise values are known only for small values of n. For example, it is known that $S(0) = 0$, $S(1) = 1$, $S(2) = 6$, $S(3) = 21$ and $S(4) = 107$, but $S(n)$ is not known for $n > 4$. This is so because $S(n)$ grows very rapidly; it is known that $S(5) \geq 47,176,870$ and $S(6) \geq 2.510^{2879}$. Thus, even assuming that we could find $S(n)$ for arbitrary n, the practicability of such testing would be at least dubious. Additionally, we do not know precise values or upper bounds of both functions for reasonable values of n, say $20 - 100$.

We can generalize this observation as follows.

REMARK 6.4 *TM-unsolvable problems have uncomputable time and space complexities.*

Otherwise, the known upper limit of time complexity (the number of steps) or the upper limit on space complexity (TM tape cells used) could be used to solve the halting problem of a UTM, to disprove the Rice Theorem, to disprove the unsolvability of the Post Correspondence Problem, and so on.

THEOREM 6.5 (ON UNSOLVABILITY OF ECP) *The Economy Collapse Problem is Turing uncomputable.*

***Proof**: By reduction of the BBP TM computing the max-shift function $S(n)$ from the Busy Beaver Problem to the ECP TM. The reduction is straightforward: The number of states in BBP TM and ECP TM are the same and equal $n = 0, 1, 2, ...$ If BBP TM accepts its empty input and halts, then ECP TM halts and the economy collapses after a finite number of steps. If BBP TM does not accept its input, i.e., does not halt, then ECP TM does not halt either. Thus ECP is recursively enumerable but not recursive, i.e., uncomputable (semi-decidable).*

We can generalize BBP to a TM starting with a non-empty tape. We will consider the latter generalization.

DEFINITION 6.6 (THE NON-EMPTY-TAPE BUSY BEAVER PROBLEM (NBBP)) *Compute functions $\Sigma(n)$ and $S(n)$ for Busy Beaver TMs starting with nonempty initial tapes.*

Let's recall another auxiliary notion. It is known that the language L_{ne} of codes of TMs accepting at least one string is recursively enumerable but not recursive, i.e., the halting problem of UTM can be reduced to L_{ne}. On the other hand, the language L_e of codes of TMs that do not accept any word is not recursively enumerable. Note that L_e is the complement of L_{ne}.

THEOREM 6.7 (ON UNSOLVABILITY OF NBBP) *The Non-Empty-Tape Busy Beaver Problem is Turing-uncomputable.*

***Proof**: By reduction of language L_{ne} to the NBBP. If TM computing L_{ne} accepts word w, then NBBP TM computes $\Sigma(n)$ and $S(n)$ and halts. If TM computing L_{ne} does not accept, then NBBP TM does not halt either.*

Now we are ready to prove the Generalized Economy Collapse Problem.

THEOREM 6.8 (ON UNSOLVABILITY OF GECP) *The Generalized Economy Collapse Problem is Turing-uncomputable.*

***Proof**: By reduction of NBBP to $GECP$.*

6.2 Economies that Never Collapse

Now, we are ready to define the *Economy Immortality Problem*, which is even more interesting than finding the maximum time that an economic system has before it collapses. Unfortunately, and not surprisingly, immortality of an economic system is not easier to achieve/decide than the maximum possible survivability of the economic system.

While there are trivial non-halting n-state machines for any given n, we assume that constraints on real economic systems will rule out such trivial, simple non-halting machines. Or in other words, we could engineer economies which were guaranteed to never collapse, but those trivial economies wouldn't be very useful or practical. On the other hand, there are infinitely many non-trivial TMs/economies that never halt. However, finding them is not easier than finding halting TMs. The problem is also unsolvable; even worse, it is non-recursively enumerable.

DEFINITION 6.9 (THE ECONOMY IMMORTALITY PROBLEM (EIP)) *Let EIP represent economies that never collapse, i.e., are immortal. For arbitrary values of n, decide whether any economy with n states is immortal.*

We observe that the Economy Immortality Problem is the complement of the Extended Generalized Economy Collapse Problem (EGECP), where we want to prove that an economy will collapse. We can prove that EGECP is Turing-unsolvable too (by reduction from L_{ne}). This implies immediately that EIP as the complement of EGECP is non-recursively enumerable (another proof can be obtained directly by reduction from L_e). This leads us immediately to the following theorem.

THEOREM 6.10 (ON UNDECIDABILITY OF EIP) *The Economy Immortality Problem is Turing-uncomputable.*

Note that EIP TMs (i.e., immortal economies) describe systems that never terminate; thus they violate the classical definition of algorithms, and they, together with operating systems and client/server programs, belong to the class of reactive systems. And reactive systems are the functional examples of implemented hypercomputational systems (claimed by many researchers to be non-implementable).

6.3 On Undecidability of Economy

As a consequence of EPP and ECP/GECP/EIP unsolvability (being some of many economic problems), we conclude:

COROLLARY 6.11 (ON UNSOLVABILITY OF ECONOMY) *Economy is TM-undecidable.*

REMARK 6.12 *This seems to be a rather pessimistic result, but who cares? Mathematics and computer science have also been proven to be TM-unsolvable; nevertheless, this did not stop fruitful research on their decidable sub-areas. However, one has to be very careful for which questions we can have a definite answer and for which we can decide only some specific cases. We cannot even conclude from above that state-controlled economies lead to undecidable economies, because some of their special cases can be decidable.*

6.4 An Extended Note on Decidable Economies

The simplest example of decidable ECP problems are economies where the Turing machine $\mathcal{M}$ output by $\mathbf{F}_{econ}$ has read/write heads allowed to move only to the right or to the left in each transition. Turing machines whose read/write heads move only to the right (or only to the left) are equivalent to push-down automata, and their accepted languages are decidable (see, e.g. [16]). This means that if the collapse/halting state exists, it will be reached after a finite number of moves of the underlying economy. We can try to interpret the meaning of economies where the TM head is allowed to move only to the right. One possible interpretation would be that each transition is associated with the unidirectional flow of time, and we are not allowed to reinterpret or correct past decisions.

Another interesting example would be the case of economies which collapse because their reusable resources (e.g., capital) ceased to be available.

A special case of decidable EIP would be modeling problems that avoid collapse of the economy caused by deadlocks when multiple clients compete for shared resources/bank capital. This includes the algorithmic solution for the Banker Problem by Dijkstra's famous Banker's Algorithm and its unsolvable generalizations. On the other hand, deadlock detection and recovery would cover the case of decidable ECP with soft collapses (soft: pending we can recover from the deadlock).

In 1965, Edsger Dijkstra [13] posed and solved (by providing a deadlock-free simple solution) the Banker Problem.

DEFINITION 6.13 (THE BANKER PROBLEM (BP)) *The Banker Problem is the abstraction of the allocation of non-shareable and reusable resources in an operating system. A banker is supposed to give loans to customers of his bank, who may apply for them as many times as they please, requesting in each application no more than the whole capital of the banker, and committing themselves to give back the loan and not apply for new loans before having returned the previous one. The banker wants to avoid any deadlock (to have money for a new client/loan) or starvation, and to spread his money as much as possible.*

Dijkstra's solution for the Banker Problem is by produced by avoiding *unsafe states* that might potentially lead to deadlocks. If the state will always lead to a deadlock independently of the lending order, it will be unsafe;otherwise it will be safe. The Banker's Algorithm assumes single-type or multiple-type non-shareable and reusable resources. In the BP problem (accidently, having nothing to do with the oil spill in the Gulf), instead of operating systems, we return to the original interpretation, that is, the bank and its generalization: the whole economy.

The economic crisis of 2008 resembles the typical scenario taken directly from the Banker's Algorithm: The banks and the world economy were in danger of immediate and unavoidable collapse (the US and world economy entered an *unsafe state* leading with almost certainty to a deadlock: halting of the whole US/world economy), because of the evaporation of capital resources.[§§] The solution provided by the US government was different from Dijkstra's recommendation of avoiding unsafe states, but relied on replenishing lost resources, i.e., the famous $700bn bailout by borrowing money mainly from China, Japan, and Saudi Arabia.[||||]

The Banker Problem seems to capture well, at the abstract level (in a way roughly parallel to dining philosphers, readers/writers, or producers/consumers, problems successfully used in concurrency and computer science from the 1960s), the danger of economic collapse caused by a lack of capital resources. Those inclined to resist formalization may well maintain that the Banker Problem is a poor analog for the financial crisis, or even for merely the part of it limited to credit freeze. But in a similar manner, one could complain that the dining-philopshers problem did not capture in all details synchronized access to i/o devices, readers/writers were not a good analog of databases, and producers/consumers did not capture properly sending messages in the internet. Nevertheless, these abstract models, by their simplicity and concentration on the most important aspects only, allowed us to solve many practical problems that initially seemed impossible to be solved properly. It does not matter that banks don't lend only their capital, and most of what they lend is what they borrowed from depositors. Simply, the banker's resources will include the banker's own capital as well as money borrowed from depositors, or received from the government (perhaps, by the bailout).

Some have claimed to us that the freeze happened because there was a sudden across-the-board change in institutions' willingness to trust the future capabilities of counterparties to repay. It has been claimed further that there is no way to build a computational account of this without making mental states of belief about other agents' prospects a fundamental part of the story. In other words, as the story here goes, the resources existed, but the bankers did not believe that their clients would repay their loans, and hence stopped lending almost completely. But beliefs can be incorporated directly into the notions of "safe" states. Unsafe states will simply be the states where we cannot get money (either because the banker does not have enough capital or does not trust his/her clients' ability to repay). It does not matter that the resource existed if the client could not get it. For the client, the bare effect is that the resource stopped being available for whatever reasons. If we wish to make life and economies more complicated, we can include a recursive (perhaps, infinite) chain of "mental" states of the type: "I believe that you believe that he believes that the society stopped to believe but perhaps would believe if . . . ." Unfortunately (or, perhaps, fortunately), all of the above has been anticipated by Dijkstra's "simple" BP/BA, and of course our own calculi (e.g., aforementioned $\mathcal{ECEC}$) are tailor-designed to represent iterated beliefs).[##]

There is however a much more severe problem with the Banker's Algorithm. As is pointed out by Tanenbaum

[§§] In truth, they changed only their owners, but the bare effect seemed like they disappeared. This is confirmed by the well-known case of John Paulson, a hedge-fund manager with his $20 billion "insurance" against the housing bubble, i.e., at least we know how to account for 20 billion of the losses. Another, less likely, option was that the resources never existed in the first instance, i.e., were "virtual" and losses and gains were also virtual, thus truly the danger of deadlock was also "virtual."

[||||] Unfortunately, we cannot ask Edgser Dijkstra whether he would recognize the bailout as a deadlock-free alternative solution to the Banker's Problem.

[##] In writing this section, we benefited from insightful reactions we received from the National Science Foundation on some of our related work.

[45], the Banker's Algorithm, although wonderful in theory, in practice (e.g., to use it as a scheduler in an operating system) is essentially useless, because it requires advance knowledge of required resources, and assumes that the number of processes and the amount of resources are static. In practice, the required resources are either unknown or difficult to estimate, and the number of processes and the amount of resources are highly dynamic. On the other hand, Silberschatz [39] claims that the centralized Banker's Algorithm can be easily extended to be used in a distributed system by designating one of the processes (the banker) as the process that maintains the information necessary to carry the Banker's Algorithm. However, he acknowledges that every resource request must be channeled through the banker, and this may require too much overhead. As we know, the capital resources in an economy are typically very dynamic and distributed.

The question is: Was the bailout the only available solution to prevent an economic freeze? We already know that the answer to this question is negative: the Banker's Algorithm provides an alternative solution (and perhaps the only real one). Did the bailout really provide a solution or only delay the time of deadlock/collapse? Can we somehow extend the Banker's Algorithm not to be used as the loan decision scheduler, but at least to send warning signals to the banker? If one is to agree that the deadlock avoidance is impossible in practice, is deadlock prevention by attacking circular-wait condition somehow possible (e.g., using semaphores, monitors or message-passing)? In other words, can we design a dynamic and distributed extension of the Banker's Algorithm for either signaling and warning of approaching deadlocks (unsafe states) or preventing deadlocks all together? Obviously, there is much opportunity for future work.

7 CONCLUSION AND NEXT STEPS

We conclude that, yes indeed, perhaps economics is partially hypercomputational in nature. A bit more precisely, there are two overarching reasons for this conclusion, both of which we have seen in play in the foregoing. First, doing economics, represented by the activity of scientist $\mathbf{F}_{econ}$, might well require hypercomputation; the reason in a nutshell is that, as Figure 3 indicates, and as material coming after that figure fleshes out,$\mathbf{F}_{econ}$ may need to ascertain, repeatedly, the answer to a Turing-undecidable question (provability). Second, economics, in seeking to provide information-processing machinery that can predict future states of relevant markets, may well need to provide machinery with the ability to surmount the undecidability that more expressive logics like $\mathcal{ECEC}$ inherit from first-order logic.

Our conclusion may not be accepted by all readers, and we assume some will be skeptics chiefly because either our modeling is perceived as perversely liberal in the direction of hypercomputation, or because this modeling is unnatural, or both. In future work, greater detail will be provided (e.g., we can provide formal analysis and theorems about undecidability in connection with mental logics). Of course, this additional detail will not address something we must admit, viz., that we don't have a rigorous set of desiderata which a formal science-of-science of the targeted economic phenomena must satisfy. On the other hand, we are not aware of the existence of such desiderata, and therefore not aware of such desiderata that would exclude the possibility of hypercomputation being part of the nature of the model. In fact, even conservative paradigms for a formal science of science, such as that presented in (see e.g. Chapter 11 of [20]) and taken whole cloth therefrom, explicitly entertain the possibility that scientists harness hypercomputation (via old-style oracles, rather than contemporary hypercomputing devices detailed in the formal literature) to produce what they produce.

REFERENCES

[1] K. Arkoudas and S. Bringsjord. (2009). Propositional attitudes and causation. *International Journal of Software and Informatics*, 3(1):47–65.

[2] R. J. Aumann. (1995). Backward induction and common knowledge of rationality. *Games and Economic Behavior*, 8:6–19.

[3] D. Balkenborg and E. Winter. (1997). A necessary and sufficient epistemic condition for playing backward induction. *Journal of Mathematical Economics*, 27:325–345.

[4] J. Barwise and J. Etchemendy. (1999). *Language, Proof, and Logic*. Seven Bridges, New York, NY.

[5] B. Bernanke. (2000). *Essays on the great depression*. Princeton Univ Pr.

[6] M. Braine. (1998). Steps toward a mental predicate-logic. In M. Braine and D. O'Brien, editors, *Mental Logic*, pages 273–331. Lawrence Erlbaum Associates, Mahwah, NJ.

[7] S. Bringsjord. (1992). *What Robots Can and Can't Be*. Kluwer, Dordrecht, The Netherlands.

[8] S. Bringsjord and M. Zenzen. (2003). *Superminds: People Harness Hypercomputation, and More*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

[9] S. Bringsjord and K. Arkoudas. (2004). The modal argument for hypercomputing minds. *Theoretical Computer Science*, 317:167–190.

[10] S. Bringsjord, J. Taylor, A. Shilliday, M. Clark, and K. Arkoudas. (July 21 2008). Slate: An Argument-Centered Intelligent Assistant to Human Reasoners. In F. Grasso, N. Green, R. Kibble, and C. Reed, editors, *Proceedings of the 8th International Workshop on Computational Models of Natural Argument (CMNA 8)*, pages 1–10, Patras, Greece.

[11] R. Carnap. (1995). *Introduction to the Philosophy of Science*. Dover, Mineola, NY.

[12] A. Cottrell, P. Cockshott, and G. J. Michaelson. (2009). Is economic planning hypercomputational? The argument from Cantor diagonalisation. *International Journal of Unconventional Computing*, 5(3-4):223–236.

[13] E. Dijkstra. (1965). Cooperating sequential processes. *Programming Languages*.

[14] C. Duhigg, (March 2008). Depression, You Say? Check Those Safety Nets. New York Times.

[15] H. D. Ebbinghaus, J. Flum, and W. Thomas. (1994). *Mathematical Logic (second edition)*. Springer-Verlag, New York, NY.

[16] J. E. Hopcroft, R. Motwani, and J. D. Ullman. (2007). *Introduction to automata theory, languages, and computation*. Prentice Hall, 3rd edition.

[17] M. Hutter. (2001). Towards a universal theory of artificial intelligence based on algorithmic probability and sequential decisions. *Machine Learning: ECML 2001*, pages 226–238.

[18] M. Hutter. (2005). *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*. Springer, New York, NY.

[19] M. Hutter. (2007). Universal algorithmic intelligence: A mathematical top→down approach. In B. Goertzel and C. Pennachin, editors, *Artificial General Intelligence*, Cognitive Technologies, pages 227–290. Springer, Berlin.

[20] S. Jain, D. Osherson, J. Royer, and A. Sharma. (1999). *Systems that learn*. MIT press.

[21] R. Jeffrey. (1964). *The Logic of Decision*. University of Chicago Press, Chicago, IL.

[22] P. N. Johnson-Laird. (1983). *Mental Models*. Harvard University Press, Cambridge, MA.

[23] J. Joyce. (1999). *The Foundations of Causal Decision Theory*. Cambridge University Press, Cambridge, UK.

[24] D. Kreps and R. Wilson. (1982). Reputation and imperfect information. *Journal of Economic Theory*, 27(2):253–279.

[25] P. Krugman, (March 2009). The Great Recession versus the Great Depression. New York Times.

[26] O. Lange. (1964). *On the Economic Theory of Socialism*. McGraw Hill, New York, NY. Originally published in 1936.

[27] S. Legg and M. Hutter. (2007). Universal intelligence: A definition of machine intelligence. *Minds and Machines*, 17(4):391–444.

[28] N. Mankiw. (2008). *Principles of macroeconomics*. South-Western Pub.

[29] R. Nozick. (1970). Newcomb's problem and two principles of choice. In N. Rescher, editor, *Essays in Honor of Carl G. Hempel*, pages 114–146. Humanities Press, Highlands, NJ.

[30] M. Osborne and A. Rubinstein. (1994). *A Course in Game Theory*. MIT Press, Cambridge, MA.

[31] J. Pollock. (1995). *Cognitive Carpentry: A Blueprint for How to Build a Person*. MIT Press, Cambridge, MA.

[32] M. Pour-El and I. Richards. (1981). The wave equation with computable initial data such that its unique solution is not computable. *Advances in Mathematics*, 39:215–239.

[33] T. Rado. (1963). On non-computable functions. *Bell System Technical Journal*, 41:877–884.

[34] L. Rips. (1994). *The Psychology of Proof*. MIT Press, Cambridge, MA.

[35] C. Romer. (2004). The Great Depression. *Britannica Encyclopedia*.

[36] S. Russell and P. Norvig. (2002). *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.

[37] L. Savage. (1954). *The Foundations of Statistics*. Dover, New York, NY.

[38] R. Selten. (1978). The chain store paradox. *Theory and Decision*, 9:127–159.

[39] A. Silberschatz, P. Galvin, and G. Gagne. (2003). *Operating system concepts*. John Wiley & Sons, Inc. New York, NY, USA, 6th edition.

[40] P. Smith. (2007). *An Introduction to Gödel's Theorems*. Cambridge University Press, Cambridge, UK.

[41] R. Solomonoff. (1978). Complexity-based induction systems: comparisons and convergence theorems. *IEEE transactions on Information Theory*, 24(4):422–432.

[42] K. E. Stanovich and R. F. West. (2000). Individual differences in reasoning: Implications for the rationality debate. *Behavioral and Brain Sciences*, 23(5):645–665.

[43] P. Suppes. (1972). *Axiomatic Set Theory*. Dover Publications, New York, NY.

[44] R. Sutton and A. Barto. (1998). *Reinforcement learning*. MIT Press.

[45] A. Tanenbaum. (2008). *Modern Operating Systems*. Prentice Hall Englewood Cliffs, NJ.

[46] W. Tracy. (2008). Paradox lost: The evolution of strategies in selten's chain store game. In *Three Essays on Firm Learning*. This is a dissertation at UCLA in the Anderson School of Management.

[47] Wikipedia, (2010). Economic collapse — Wikipedia, the free encyclopedia. [Online; accessed 22-September-2010].

[48] A. Wiles. (1995). Modular elliptic curves and Fermat's last theorem. *Annals of Mathematics*, 141(3):443–551.

[49] A. Wiles and R. Taylor. (1995). Ring-theoretic properties of certain Hecke algebras. *Annals of Mathematics*, 141(3):553–572.

[50] Y. Yang, M. Braine, and D. O'Brien. (1998). Some empirical justification of one predicate-logic model. In M. Braine and D. O'Brien, editors, *Mental Logic*, pages 333–365. Lawrence Erlbaum Associates, Mahwah, NJ.

[51] Y. Yang. (2006). Toward a mental decision logic of the small-grand problem: Its decision structure and arithmetization. In *Proceedings of the Twenty-Eighth Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum Associates, Mahwah, NJ.