On Logicist Agent-Based Economics*

Selmer Bringsjord[1,2,3] • John Licato[1] • Atriya Sen[1]
Joe Johnson[1] • Alexander Bringsjord[4] • Joshua Taylor[6]

Dept of Computer Science[1] • Dept of Cognitive Science[2] • Lally School of Management & Technology[3]
Rensselaer Polytechnic Institute (RPI) • Troy NY 12180 USA

PricewaterhouseCoopers LLP[4] • 300 Atlantic Street • Stamford CT 06901 USA

Siege Technologies[6] • 540 North Commercial Street • Manchester NH 03101 USA

Abstract.
We share herein information about our logicist agent-based approach to economics. This approach is: rooted in a space ($\mathscr{CC}$) of singularly expressive computational logics (**cognitive calculi**) that enable fine-grained modeling of cognitive agents; verifiable; and simulation-based. Overall, we seek to provide (to humans) intelligent artificial agents able to make discoveries about, and explain and predict the future of, interacting agents (whether human or artificial), where these interacting agents are in our approach modeled and simulated by taking full, explicit account of all the main elements of cognition (i.e., belief, knowledge, perception, communication, goals, etc.). We present two case studies designed to convey a more concrete sense of our approach. One case study involves the "chain-store paradox," and the other involves the devious reasoning of someone planning to establish a Ponzi scheme designed to evade detection by authorities. Both case studies are simpler than the kind of communication-rich, massive, multi-agent micro- and macro-level simulations we strive for as our approach matures, but they convey the gist of our approach.

1 Introduction

The plan of the paper is as follows. We first encapsulate our basic model of an economist as one who produces artificial agents capable of issuing valid fiscal predictions, as well as, when appropriate, explanations and discoveries (§2). Next, we briefly explain that our brand of economics is: rooted in logicist AI; proof-based (and hence verifiable); simulation-based; and committed to modeling down to *all* the main elements of cognition at the individual-agent level (§3). Our new paradigm is logicist for two overarching reasons: One, in this paradigm, the (idealized) practice of economics is an activity having predictive, explanatory, and discovery power on the strength of using expressive, robust computational logics (which are members of our space of **cognitive caculi** $\mathscr{CC}$) to assess and confirm hypotheses. Two, the use of these calculi includes the modeling of agents in order to achieve the predictive power in question. In this paradigm, the full-range of the cognition that distinguishes *being a person* must be formalized: belief, knowledge, perception, reasoning, decision-making, communication, reading, goals, planning, emotion, etc. [1]. As such, 'agent' for us is a term roughly in line with the most sophisticated intelligent artificial agents defined in AI, e.g. the agents defined in [2]. We maintain that all the main elements of cognition at the individual level are directly relevant to economic behavior of individuals, groups, nations, and groups of nations; and that ultimately the only way to achieve true precision and power in economics is to harness technology able to model and reason over all the main elements of cognition of individuals.

Predictions in our paradigm are issued on the strength of simulations generated by formal reasoning; the simulations in question are verifiable because the reasoning in question, consisting of proofs and formal arguments, are machine-checkable, and the code needed for proof/argument verification is tiny and can be formally verified by traditional techniques.[1] In the present paper, we use a very robust cognitive calculus [$\mathcal{D}^y\mathcal{CEC} \in \mathscr{CC}$, which alone far exceeds so-called "BDI logics"; for details, see e.g.

* We are deeply grateful for penetrating comments received from three anonymous reviewers, which led to much deep contexualization of our paradigm. And we are indebted to Naveen Sundar Govindarajulu of Yahoo for substantial work carried out while in Bringsjord's RAIR Lab.

[1] This overall approach is laid out and explained quasi-formally in [3]. For a quick introduction to standard program verification, see [4].

[5,6,7], and we take up this issue again briefly in §6] allowing for explicit representation of the internal structure of multiple, interacting agents.[2] Next, with our recommended form of economics in mind, we carry out two case studies designed to illustrate, at least to a suggestive degree, logicist agent-based economics. One case study is based on the famous Chain-Store Paradox (§4); the other on a Ponzi scheme scenario. (More elaborate modeling and simulation in our paradigm is carried out in [5].) In §6 and §7 we discuss related work, and rebut a series of objections, respectively. We conclude with brief remarks regarding future work (§8).

2 Our Abstract Science-of-Science Framework (High-Level)

2.1 The Generic Human-Agent Model

At least from the standpoint of formal logic, the goal of a rational scientific enterprise like economics can be conceived as finding *models* that fit some observed data expressed as formulae in one or more formal logical languages, with the hope that these models eventually accurately describe the underlying reality from which these data flow. The formal-and-computational study of the scientific process is what we call *the formal science of science*. Because economics, unlike physics, has a number of competiing paradigms, we believe it is very important to have some kind of rigorous meta-scientific framework to judge paradigms.

An introduction to a particularly promising computational science of science can be found in [9], which is recursion-theoretic and hence based on mathematical logic. One of the first scientific paradigms considered in [9] is *language identification*, which involves two entities, Nature and a Scientist.[3] A language L here is taken to be the familiar 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.[4] Only the recursively enumerable languages $\mathcal{E}$ are considered in [9].[5] Our Scientist $\mathbf{F}$ operates in some fixed and unknown reality L_t, and perceives 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. For us, given our computational orientation, it is convenient that these hypotheses are in fact formally defined information-processing machines (see below).

The Scientist is more specifically 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 corresponding to a representation of some partial-recursive function in some *programming system* ν. The 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 examining the data it has seen from Nature so far. Figure 1a illustrates this process.

In our opinion, this framework conspicuously lacks some elements which are ubiquitous in formal science.[6] One missing aspect is the concept of a proof (or at least a proof-sketch or rigorous argument) 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 1b contains an overview of one such possible improved framework. As should be clear from this figure, the basic new idea is that the 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,[7] as confirmed

[2] This calculus is not to be confused with the *deontic*, as opposed to the — used herein — *dynamic* calculus. The former subsumes a dyadic deontic logic. E.g., see [8].

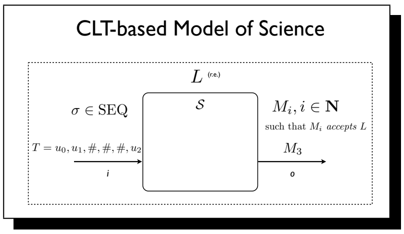
[3] The notion of equating science with language identification is analogous to equating computation with language recognition.

[4] 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.

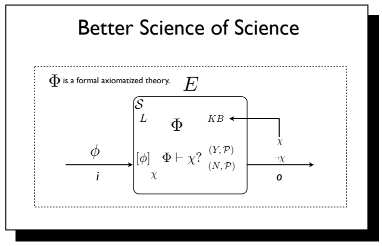
[5] This is of course a presupposition against hypercomputation (information-processing above Turing machines in the Arithmetic Hierarchy), which we find objectionable [10]. But nothing hinges on this issue in the present paper.

[6] The two best formal scientists of science, Suppes [11] and the great logician Tarski [12], would agree.

[7] Here 'theory,' in keeping with formal logic, is an axiom system; see e.g. [13].



(a) Conventional



(b) Logcist

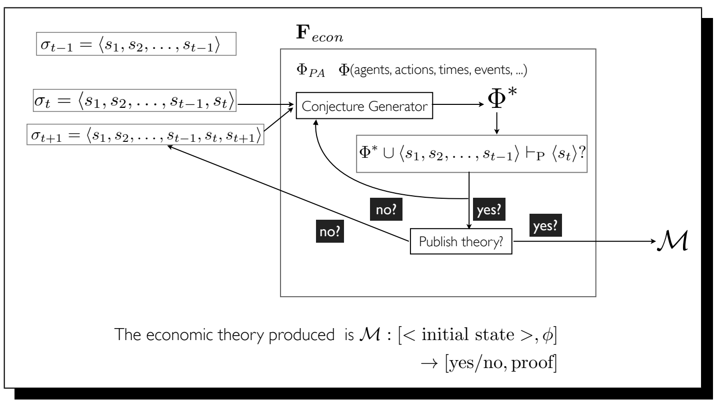
Fig. 1: Two CLT-Based Formal Science-of-Science Views

by some proof $\mathcal{P}$. If the answer is in the affirmative, χ is added to the knowledge-base for the science in question.[8]

2.2 Applying the Generic Model to Economics

Our logicist science-of-economics framework $\mathbf{F}_{econ}$ is shown pictorially in Figure 2. In this framework, the Scientist $\mathbf{F}_{econ}$ produces an information-processing machine $\mathcal{M}$ (which needn't be a Turing machine or an equivalent thereof), or in AI terms, an *intelligent agent* [14], that is capable of taking in a set of initial or prior states of the relevant environment or market, and of answering whether a certain future state-of-affairs ϕ is presently true, or 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 environments. (We write $\langle s_i \rangle$ to indicate that a state has been expressed in declarative form in the form of as set of formulae.) Note that even in the general case involving only inexpressive extensional logics like first-order logic (which we far exceed below by deploying $\mathcal{D}^y\mathcal{CEC}$), 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 propose a theory Φ^* that it tests on the available data. If the conjecture agrees with the available data, the scientist then decides either to publicly assert 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, agent utterances, times, events etc.; more about this issue below. In the specific context of economics, we of course seek to produce models of $\mathbf{F}_{econ}$ and $\mathcal{M}$ for modeling specific economic phenomena.

Fig. 2: Science of Sciences applied to Economics



[8] Because of space limitations, we omit any coverage of the growing body of formal results regarding our new science-of-science model.

3 Logicist Agent-Based Economics

In our approach, $\mathbf{F}_{econ}$, when provided with temporal data from an environment, produces a logical theory Γ which is used to predict future states of that environment. The logical system in which Γ is expressed includes, necessarily, expressive intensional operators that allow representation of, and reasoning about, all of the structure of cognition of interacting individual agents, along with inanimate factors in the environment or market. Since the system is based on declarative information, theorems announced by the system are verifiable. Also, since Γ includes information about individual agents, the system can be used for simulation by deriving theorems about the temporal states of the system. This dimorphism between simulation and verification parallels and follows from *individual agent reasoning* vs. *system reasoning* in multi-agent systems. The system is cognitive, as we represent the mental states of agents in detail. Calculi used in representing Γ can be decomposed into two smaller sets of calculi:

Physical Calculi This is the part of the calculus that describes physical systems, external systems, market rules and the dynamics of inanimate systems. We generally use the *event calculus* [15] or situation calculus [16] for our physical calculus; the granularity of the former suffices for the relatively simple case studies presented below.

Cognitive Calculi Our cognitive calculi contain physical calculi as proper fragments, and should model as accurately as possible, for a given environment, agents' beliefs, knowledge, desires, intentions, communication, etc., and how these elements evolve and change. We present two cognitive calculi from $\mathscr{CC}$. The first, the aforementioned Cognitive Event Calculus ($\mathcal{CEC}$), is used in the chain-store paradox study; the second, the **Dynamic Cognitive Event Calculus** ($\mathcal{D}^y\mathcal{CEC}$), is used in the Ponzi-scheme modeling study.

For proof checking and automated theorem-proving, we use in the present paper a system built upon a denotational proof language (DPL).[9] Given a logical language $\mathcal{L}$, a DPL $\mathcal{K}$ for that language provides a way to formally express proofs in $\mathcal{L}$ in the language of $\mathcal{K}$. The syntax of a DPL is built using the $\lambda\mu$ calculus [17], whose basic syntactic categories are *propositions*, *deductions*, and *methods*. Evaluation in a DPL reduces to proof checking. A Deduction D, when evaluated in the context of an assumption base β, produces a proposition P, denoted as $\beta \vdash D \rightsquigarrow P$. Methods in the $\lambda\mu$ calculus, or $\lambda\mu$-methods, are arbitrary abstractions over deductions and are analogous to functions in the λ calculus. $\lambda\mu$-methods abstract over common reasoning scenarios and allow for modular non-monolithic automatic theorem proving. Methods can be *primitive* or *derived*. Derived methods are obtained from primitive methods through a certain set of operations which, for example, include method composition. Primitive methods cannot be decomposed any further and correspond to primitive rules of reasoning in $\mathcal{L}$.

We now present a specification of our model in terms of the logicist framework for science-of-science introduced above. *Moments* is the set of all timepoints of interest on a discrete timeline, $\mathcal{P}$ the set of all formulae in the cognitive calculus $\mathcal{L}$ ($\mathcal{CEC}$and $\mathcal{D}^y\mathcal{CEC}$ in the present paper), and $f_i(\mathcal{P})$, called **formats**, specify subsets of logical formulae we are interested in. The set of all formats is $\mathcal{F}$.

$\mathbf{F}_{econ}$ Viewed from an agent perspective, $\mathbf{F}_{econ}$ could be specified using some inductive formalism, e.g., inductive logic programming (see [18]) or probability theory axiomatized declaratively (e.g. [19]). The economist takes in a finite set of formulae representing experimental data and a finite set of formats which determine the shape of the theory we are interested in; and outputs a machine specification $\langle \mathcal{M}_i \rangle$.

$$F_{econ} : 2^{\mathcal{P}} \times 2^{\mathcal{F}} \mapsto \{\langle \mathcal{M}_i \rangle | i \in \mathbb{N}\}$$

$\mathcal{M}_{econ}$ This is defined as a *logicist simulation*: a function mapping each instant of time and experimental data represented declaratively to a list of formulae of specific *formats*.

$$\mathit{Simulation} : \mathit{Moments} \times 2^{\mathcal{P}} \mapsto f_1(\mathcal{P}) \times \ldots \times f_n(\mathcal{P})$$

To construct the above definition of a simulation, we have a $\lambda\mu$-method m_i for each format of interest f_i. Then $\mathcal{M}_{econ}$ is defined as

$$\mathcal{M}_{econ} = \{m_1, \ldots, m_n\} \text{ where } m_i : \mathit{Moments} \times 2^{\mathcal{P}} \mapsto f_i(\mathcal{P})$$

Γ_{econ} The theory is represented in $\mathcal{L}_{econ}$: $\Gamma_{econ} \subset \mathcal{P}$. Two examples, Γ_{csp} and Γ_{ponzi}, are illustrated below.

[9] DPLs were invented and introduced by Arkoudas [17].

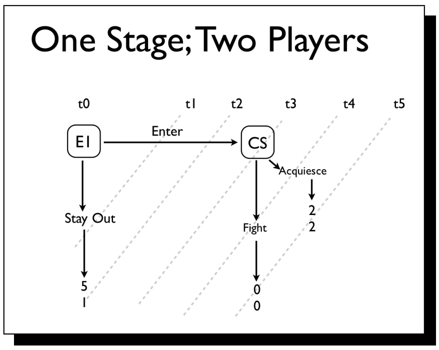
The following two case studies illustrate how one might go about specifying the details and implementing the formal specification.[10]

4 The Chain-Store Case Study

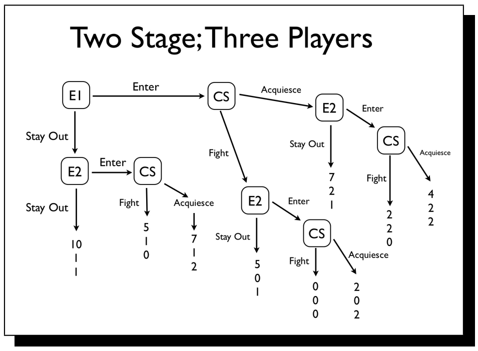
In §4.1 we introduce the Chain-Store Paradox (CSP). In §4.2 we show manually how our approach to economics works in connection with CSP. And in §4.3 we show how our approach can be automated.

4.1 Overview of the Chain-Store Paradox

Nobelist Reinhard Selten [20] introduced the remarkably fertile Chain Store Paradox (CSP), which centers around strategic interaction between a "chain store" (CS; e.g., Wal-Mart, McDonald's, or even, say, Microsoft) and those who may attempt to enter the relevant market and compete against CS. The game here is an n-stage, $n+1$ one, in which the $n+1^{th}$ player is 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; 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 here by the fact that we set the values, without loss of generality, to be, resp., 5, 2, 1, and 0. Please see Figures 3a and 3b, 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 3a: viz., that we have used diagonal dotted lines, with labels, to indicate key timepoints in the action.



(a) One-Stage, Two-Player Snapshot in CSP



(b) Two-Stage, Three-Player Snapshot in CSP

Fig. 3: Example Snapshots in CSP

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

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

In the second sense of 'paradox,' a theorem is simply regarded by many to be extremely counterintuitive, but no outright contradiction is involved. A famous example in this category is Skolem's Paradox, elegantly discussed in [13]. Finally, in the third sense of 'paradox,' a contradiction is produced,

[10] They are not complete, in a sense, as they lack an implementation or a specification of automating $\mathbf{F}_{econ}$. The formal specification also holds if one considers $\mathbf{F}_{econ}$ to be a human practitioner. The specification, if not the implementation, of the case studies can be considered complete in this latter sense.

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: [21]. It is into this third category of 'paradox' that CSP falls. More specifically, we have first the following definition and theorem.[11]

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}, ..., 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.[12] From our formal science-of-science perspective, and specifically from the perspective of the CLT-based science of sciences framework shown in Figure 2, 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`.

[11] Selten himself doesn't provide a fully explicit, verifiable proof of the theorem in question. For more formal treatments, and proofs of backward induction, see e.g.,[22,23]. In the interest of economy, we provide only a proof-sketch here, and likewise for the theorem thereafter for deterrence.

[12] Selten in [20] prophetically remarked that he never encountered someone who said "he would behave according to induction theory [were he the Chain Store]."

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 entering 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.[13] 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, since the purpose of the present paper is of course to introduce and take genuine steps toward a new approach to economics rooted in logicist agent-based AI.

4.2 Semi-Automated Analysis of the Chain-Store Paradox

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/calculi used. Our calculi use more detailed machinery than has been used before in the chain-store literature. 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 [7], which allows for epistemic and communication operators to apply to sub-formulas in full-blown quantified modal logic, in which, as we have said, the standard event calculus is encoded.

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 [26] and [27], knowledge is interpreted as a simple function, rather than as an operator that can range over very expressive formulae that carry parameters for timepoints, actions, goals, plans, and utterances. This same limited, simple treatment of intensional operators 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 interested readers to the rather intricate proof.[14] which is too large to present in the present paper. This proof is in $\mathcal{ECEC}$, implemented and machine-checked in the Slate system [28], 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 2.

4.3 Toward Automatic Analysis of the Chain-Store Paradox

We have implemented two simulations of the pair of diverging outcomes. Since reasoning in CSP involves finitely many levels of iterated belief, we include some rules of inference not present in $\mathcal{CEC}$ of [7], notably *conjunction introduction* and *modus ponens* within finitely iterated beliefs: $[\wedge I^*]$ amd $[\rightarrow E^*]$.

---

[13] 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 [25].

[14] Available for viewing at http://kryten.mm.rpi.edu/chain-store-color.pdf.

$$\frac{\mathbf{C}(P)}{\mathbf{C}(\mathbf{K}(a,P))}\ [R_0]$$

$$\frac{}{\mathbf{C}((P_1 \leftrightarrow P_2) \rightarrow (P_1 \rightarrow P_2))}\ [R_{9.1}]$$

$$\frac{}{\mathbf{C}((P_1 \leftrightarrow P_2) \rightarrow (P_2 \rightarrow P_1))}\ [R_{9.2}]$$

$$\frac{\mathbf{B}(a_1,\ldots\mathbf{B}(a_n,P_1)) \quad \mathbf{B}(a_1,\ldots\mathbf{B}(a_n,P_2))}{\mathbf{B}(a_1,\ldots\mathbf{B}(a_n,P_1 \wedge P_2))}\ [\wedge I^*]$$

$$\frac{\mathbf{B}(a_1,\ldots\mathbf{B}(a_n,P_1 \rightarrow P_2)) \quad \mathbf{B}(a_1,\ldots\mathbf{B}(a_n,P_1))}{\mathbf{B}(a_1,\ldots\mathbf{B}(a_n,P_2))}\ [\rightarrow E^*]$$

Before delving further into the formal analysis, a few notes on our logical vocabulary: $plans(a,\alpha,t)$ indicates that a plans to perform an action α at a time t, and $does(a,\alpha,t)$ that a actually does. $act(a)$ and $react(a)$ denote a's turn in the game and CS immediately following turn, resp. In terms of our formal-science-of-science framework we have:

f_i : We are interested in only one format of formulae:

$$\{\mathbf{B}(\bar{a_1},\ldots\mathbf{B}(\bar{a_n},plans(\bar{a_k},\bar{\alpha},\bar{\tau}))\ldots)|\bar{a_i} \in \mathsf{Agent} \wedge \bar{\alpha} \in \mathsf{ActionType} \wedge \bar{\tau} \in \mathsf{Moment} \wedge i \in \mathbb{N}\}$$

The number of belief iterations is variable. A formula of the above format with no iterated belief operators is just a *plans* formula.

Γ_{csp} : The theory is given by the following set of axioms (discussed below).

$$\Gamma_{csp} = \{\gamma_{plan}, \gamma_{cs\text{-}plan}, \gamma_{ent\text{-}plan}, \gamma_{cs\text{-}rat}\}$$

$\mathcal{M}_{csp}$: The $\lambda\mu$-method for the above format, $plans^{B*}_{a_k}$, comprising $\mathcal{M}_{csp}$, is discussed below.

Agents can plan multiple action types at the same moment, but only one action can happen based on some priority. For CS, the *fight* action has higher priority than the *coop*/acquiesce action. An action performed by an agent must have been planned by the agent.

$$\gamma_{plan} : \forall\, a,\alpha,t\,[does(a,\alpha,t) \rightarrow plans(a,\alpha,t)] \tag{1}$$

Planning Axioms for the Chain Store: The planning axioms for the CS are assumed to be common knowledge; there are two axioms for each action type of CS. The following axiom says that it's common knowledge that if CS plans to fight against an entrant, then CS believes that the total payoff gained if CS fights against this entrant is greater than the total payout gained by coöperating with this agent. A similar axiom governs acquiescence.

$$\gamma_{cs\text{-}plan} : \mathbf{C}\left(\forall\, e\left[\begin{array}{l} plans(\mathsf{cs}\ , fight(e), react(e)) \rightarrow \\ \mathbf{B}\left(\mathsf{cs}\ , \begin{array}{c} pay_{tot}(e, fight(e)) > \\ pay_{tot}(e, coop(e)) \end{array}\right)\end{array}\right]\right) \tag{2}$$

Planning Axioms for the Entrants: It is assumed that it is common knowledge that if an entrant plans to enter, then it believes that CS will coöperate.

$$\gamma_{ent\text{-}plan} : \mathbf{C}\left(\forall\, e\left[\begin{array}{l} plans(e, enter, act(e)) \rightarrow \\ \mathbf{B}(e, plans(\mathsf{cs}\ , coop(e), react(e)))\end{array}\right]\right) \tag{3}$$

Rationality Axioms for the Chain Store: In order to obtain backward-induction (BI) behavior, we assume all agents have common knowledge of rationality. [29] shows that if in games of perfect information common knowledge of rationality is present, the BI argument holds in that game. We assume that it is common knowledge that CS is a rational planner (but not necessarily a rational actor; and hence CS is defined in terms of *plans* rather than *does*). It is common knowledge that if all later entrants have planned to enter, the total payout for CS is better if it coöperates with the current entrant. There is one such axiom for each entrant. The following axiom is about the CS's rationality for the first entrant when there are only three entrants in the game.

$$\gamma_{cs\text{-}rat} : \mathbf{C}\left(\begin{array}{l} \mathit{plans}(e_2, \mathit{enter}, \mathit{act}(e_2)) \wedge \\ \mathit{plans}(e_3, \mathit{enter}, \mathit{act}(e_3)) \wedge \\ \left(\begin{array}{l} \mathit{pay}_{tot}(e_1, \mathit{fight}(e_1)) > \\ \mathit{pay}_{tot}(e_1, \mathit{coop}(e_1)) \end{array}\right) \end{array}\right) \tag{4}$$

For a given number of entrants, we have *$\lambda\mu$-generators* for automatically producing $\lambda\mu$-methods for mimicking the reasoning carried out by individual entrants and the chain store. Specifically, the method $\mathit{plans}^{B*}_{a_k}$ for agent a_k takes in as arguments a list of agents and proves an iterated belief statement about the plan for a_k. In the $\lambda\mu$ calculus notation, with β denoting the assumption base of the axioms needed for the simulation:

$$\beta \vdash \mathit{plans}^{B*}_{a_k}(a_1, \ldots, a_n) \rightsquigarrow \mathbf{B}(a_1, \ldots \mathbf{B}(a_n, \mathit{plans}(a_k, \alpha, \tau)) \ldots)$$

The output of the simulation is obtained by running these methods on an empty list of agents:

$$\beta \vdash \mathit{plans}^{B*}_{a_k}() \rightsquigarrow \mathit{plans}(a_k, \alpha, \tau)$$

The run times and the number of $\lambda\mu$-method calls for simulating the BI argument for varying numbers of entrants are shown in Table 1.

Though theoretically appealing, the outcome of BI argumentation is not very realistic for many reasons, foremost among which is the opposing result based on deterrence. In the next section we introduce constructs needed in an entrant agent for deterrence to be successful, mainly learning/generalization and a percept operator which allows us to model agents which observe their environment. Before that we demonstrate a small scenario in which CS "steps over" the BI argument and plans to fight against entrants to deter them. This can occur if CS carries out the BI argument and decides on the basis of the argument to fight and deter. We assume that the following axiom, *deterrence*, for deterrence, holds. The axiom say that if for each entrant CS believes that entrant plans to enter, then CS will plan to fight (possibly in the hope of deterring future entrants).

$$\wedge_{i=1}^{n} \mathbf{B}(\mathsf{cs}\ , \mathit{plans}(e_i, \mathit{enter}, \mathit{act}(e_i))) \rightarrow \wedge_{i=1}^{n} \mathit{plans}(\mathsf{cs}\ , \mathit{fight}(e_1), \mathit{react}(e_1)) \tag{5}$$

The deterrence action "over stepping" the BI argument is then obtained by the following proof, which uses methods defined for the BI argument.

$$\beta \vdash \left[\begin{array}{l} \textbf{conjoin}\ \mathit{plans}^{B*}_{a_1}(\mathsf{cs}\ ), \ldots, \mathit{plans}^{B*}_{a_n}(\mathsf{cs}\ )); \\ \textbf{modus-ponens}\ \mathit{deterrence},\ \wedge_{i=1}^{n} \mathit{plans}^{B*}_{a_i}(\mathsf{cs}\ ) \end{array}\right] \rightsquigarrow \wedge_{i=1}^{n} \mathit{plans}(\mathsf{cs}\ , \mathit{fight}(e_i), \mathit{react}(e_i)) \tag{6}$$

On Implementing $\lambda\mu$-Methods: A simulation of agents in the chain-store scenario depends on agents perceiving the actions of other agents through time. However, many of the axioms and inference rules that describe agents' behaviors are based on an agent knowledge. It will be very common, then, as a result, that certain deductive procedures will depend on the presence of formulae of the form $\mathbf{K}(a, P)$, indicating that an agent a knows some proposition P; but the simulation will have only asserted that $\mathbf{P}(a, P)$, i.e., that a perceived P. A useful deductive method, then, takes a list of formulae from the assumption base of the form $\mathbf{P}(a, P)$ and derives (through the use of rule DR_4 of the $\mathcal{CEC}$) the formula $\mathbf{K}(a, P)$, and then invokes another deductive procedure *within the scope* of these additional results:

```
(define-method invoke-with-perceived-knowledge (perceptions method)
  (if (endp perceptions) (! method)
    (dseq
      (!'dr4 (first perceptions))
      (!'invoke-with-perceived-knowledge (rest perceptions) method))))
```

The following formula encodes the definition of 2-learning in the $\mathcal{CEC}$. It says that an agent l learns if and only if when l knows that an agent a does some action α at distinct moments t and t', then l believes that, at any subsequent moment t'', a will again perform α.

$$\forall l \left[\begin{array}{c} \mathit{learns}(l) \leftrightarrow \\ \forall a, \alpha, t, t' \left(\begin{array}{l} \left(\begin{array}{l} \mathbf{K}(l, \mathit{does}(a, \alpha, t)) \wedge \\ \mathbf{K}(l, \mathit{does}(a, \alpha, t')) \wedge \\ t \neq t' \end{array} \right) \rightarrow \\ \forall t'' \left(\begin{array}{l} t < t' < t'' \rightarrow \\ \mathbf{B}(l, \mathit{does}(a, \alpha, t'')) \end{array} \right) \end{array} \right) \end{array} \right]$$

A common pattern is to show, provided that an agent learns, and given some of its relevant perceptions, that the agent believes that another agent will do some particular action at a subsequent moment.

The $\lambda\mu$-method, **learning**, takes as arguments two agents, a_1 and a_2, an action, α_1, and three moments, t_1, t_2, and t_3. It specializes the learning definition with the first agent, obtains the right-hand side of the resulting biconditional, and specializes that to produce a conditional that claims that if a_1 knows that a_2 did α_1 at t_1, and t_2 and t_1 and t_2 are distinct, then a_1 believes that a_2 will do α_1 at any subsequent moment. An automated reasoner, in this case SNARK [30], is invoked with this result, the prerequisite knowledge, and the axioms describing time. SNARK, as only a first-order system, sees the knowledge and belief formulae only as opaque propositions, but can perform the necessary reasoning concerning moments, to prove the universally quantified conditional $\forall t'' t_1 < t_2 < t'' \rightarrow \mathbf{B}(a_1, \mathit{does}(a_2, \alpha_1, t''))$ which is then specialized with t_3 to produce $t_1 < t_2 < t_3 \rightarrow \mathbf{B}(a_1, \mathit{does}(a_2, \alpha_1, t_3))$ (which SNARK cannot confirm directly, since $\mathbf{B}(a_1, \dots)$ appears as an opaque proposition). SNARK can be used, however, to then prove that $t_1 < t_2 < t_3$, and that as a result, the consequent $\mathbf{B}(a_1, \mathit{does}(a_2, \alpha, t_3)$ holds. The success of **learning** as applied to a_1, a_2, α, and t_1, $\dots$, t_3, depends on the following formulae being present in the assumption base: (i) the definition of learning; (ii) that a_1 is a learner; (iii) that a_1 knows that a_2 did α_1 at t_1 and t_2; and (iv) the time axioms (used to confirm that $t_1 < t_2 < t_3$). The definition of **learning** follows:

```
(define-method learning (agent1 agent2 action time1 time2 future)
  (dlet* ((x (!'uspec*
               (!'iff-elim
                 (!'uspec (!'claim *defn-learn*) agent1)
                 (!'claim ($`(learns ,agent1))))
               (list agent2 action time1 time2)))
          (k1 (!'claim ($`(knows ,agent1
                            (does ,agent2 ,action ,time1)))))
          (k2 (!'claim ($`(knows ,agent1
                            (does ,agent2 ,action ,time2)))))
          (uc (!'snark-prove
                ($`(forall ((?ttt moment))
                     (if (prior ,time2 ?tt)
                       (believes ,agent1
                         (does ,agent2 ,action ?ttt)))))
                (list* x k1 k2 *time-axioms*)))
          (c (!'uspec uc future)))
    (!'snark-prove
      ($`(believes ,agent1 (does ,agent2 ,action ,future)))
      (list* c *time-axioms*))))
```

The simulation does not provide the knowledge necessary for **learning** to successfully prove that a_3 believes that CS will choose Fight at the moment t_8. In particular, the simulation provides only the perceptions (among others) $\mathbf{P}(a_3, \mathit{does}(\mathsf{cs}\ , \mathsf{Fight}, t_2))$ and $\mathbf{P}(a_3, \mathit{does}(\mathsf{cs}\ , \mathsf{Fight}, t_5))$. However, provided that these perceptions are in the assumption base and recorded, **invoke-with-perceived-knowledge** and **learning** can be easily combined to prove $\mathbf{B}(a_3, \mathit{does}(\mathsf{cs}\ , \mathsf{Fight}, t_8))$ by first deriving all the knowledge associated with the perceptions, and then invoking **learning**:

```
(!'invoke-with-perceived-knowledge *observations*
 (mu () (!'learning (%'a3) (%'cs) (%'fight) (%'t2) (%'t5) (%'t8))))
```

By using these types of abstractions over deductive procedures, we have automated significant portions of the reasoning that occurs in these cognitively-based simulations. This automation incorporates standard off-the-shelf reasoning components, and integrates them with $\mathcal{EC}$-specific procedures. These combinations provide powerful proof-based tools for automating computational simulation and producing verifiable results. Table 1 shows simulation/proving run times, as well the number of $\lambda\mu$-methods called to produce the proofs for the backward-induction and deterrence scenarios.

Table 1: Performance on a 2.8 GHz quad core 2 GB machine

	Backward Induction		**Deterrence**	
Entrants	Time (s)	$\lambda\mu$ calls	Time (s)	$\lambda\mu$ calls
1	0.23	262	0.26	321
2	1.12	1296	1.13	1201
3	4.60	4330	3.87	3440
4	15.76	12322	11.93	8878
5	51.03	32150	36.42	21674
6	161.70	79490	109.63	51106
7	520.72	189534	374.86	117650
8	1948.42	440346	1344.13	266098

The run times grow exponentially due to the nature of the problem; every time an entrant is added, the new entrant has to repeat the reasoning of all previous entrants in order to plan. Fortunately, $\lambda\mu$-methods allow easy parallelization via a simple blackboard-style architecture. The blackboard prover takes in a goal $\mathcal{G}$ to be proved and a set $\mathcal{M}$ of $\lambda\mu$-methods relevant to the reasoning problem represented by the goal. This prover contains a blackboard $\mathcal{B}$ containing the current goal to be proved. Initially, the goal to be proved is written on the blackboard and the prover tries all the methods until one of the methods produces a deduction of the form $\mathcal{G}' \rightarrow \mathcal{G}$. The prover then erases $\mathcal{G}$ as the current goal and writes $\mathcal{G}'$ on $\mathcal{B}$ as the new goal to be proved; it continues until it is left with no further goal, at which point it signals success.

5 Case Study II: Ponzi Scheme Modeling

In this section we show how various aspects of Ponzi schemes can be modeled in our calculi. As is well-known, the schemer collects money from investors seeking a promised high rate of return, but simply pays out on the strength of new money coming in from new investors.[15] For the dynamics of the underlying fund and the promised amount to the investors, we use the continuous model presented by [31]. In this model it is assumed that the Ponzi schemer collects K amount of cash from investors at the beginning and promises to invest at a rate of r_p, the Ponzi rate, but actually invests them at a rate r_n, the nominal rate, with $r_p > r_n$. It is assumed that the density of withdrawals is r_w, cash comes in at a rate of $s_0 e^{r_i t}$, and the initial amount of money present in the fund is C. The amount of actual funds $S_a(t)$ and the amount of promised funds $Sp(t)$ are then given by

$$
\begin{aligned}
S_a(t) &= \frac{r_w[s_0 - (r_i - r_p + r_w)K]}{(r_p - r_n - r_w)(r_i - r_p + r_w)} e^{(r_p - r_w)t} \\
&+ \frac{s_0(r_i - r_p)}{(r_i - r_n)(r_i - r_p + r_w)} e^{r_i t} + \left(C - \frac{s_0(r_n - r_p) + K r_w (r_i - r_n)}{(r_i - r_n)(r_n - r_p + r_w)} \right) e^{r_n t} \\
S_p(t) &= \frac{s_0}{r_p - r_i - r_w} \left(e^{(r_p - r_w)t} - e^{r_i t} \right) + K e^{(r_p - r_w)t}
\end{aligned}
$$

[15] One cannot by the way stipulate that these investors are hoodwinked. They may even know or at least suspect that they are handing their money to a Ponzi schemer. In more robust modeling and simulation using our tools, we differentiate investor beliefs with respect to the nature of the organization they are giving money to.

Call the axiomatization of the above dynamics Γ_{dyn}. Instead of deploying on-hand axiom systems for arithmetic (e.g., the well-known **PA** or **Q**) to simulate the dynamics, we make of use of primitive $\lambda\mu$-methods as procedural rewrite methods for functions S_a and S_t. In order to represent dynamic cognitive states, we modify $\mathcal{CEC}$ once again, this time to include time nominals for the operators for perception, belief, knowledge, and common knowledge. The syntax and the rules of $\mathcal{D}^y\mathcal{CEC}$ are shown in Figure 4. We introduce a new operator for declarative communication, $\mathbf{S}(a, b, t, \phi)$: a communicates ϕ to b at time t. This operator is based on the *Inform* speech act operator discussed in [32], and is modeled, partially, by inference rule R_{12}. Another new operator is the future *intention* operator $\mathbf{I}$, which parallels similar operators in the literature on BDI agents.

Fig. 4: Dynamic Cognitive Event Calculus

Syntax

$S ::=$ Object | Agent | Self | ActionType | Action $\sqsubseteq$ Event | Moment | Boolean | Fluent | RealTerm

$f ::=$
- *action* : Agent $\times$ ActionType $\rightarrow$ Action
- *initially* : Fluent $\rightarrow$ Boolean
- *holds* : Fluent $\times$ Moment $\rightarrow$ Boolean
- *happens* : Event $\times$ Moment $\rightarrow$ Boolean
- *clipped* : Moment $\times$ Fluent $\times$ Moment $\rightarrow$ *Boolean*
- *initiates* : Event $\times$ Fluent $\times$ Moment $\rightarrow$ Boolean
- *terminates* : Event $\times$ Fluent $\times$ Moment $\rightarrow$ Boolean
- *prior* : Moment $\times$ Moment $\rightarrow$ Boolean
- *interval* : Moment $\times$ Boolean

$t ::= x : S \mid c : S \mid f(t_1, \ldots, t_n)$

$\phi ::=$ t : Boolean $\mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid$ $\mathbf{M}(a,t,\phi)\mathbf{P}(a,t,\phi) \mid \mathbf{K}(a,t,\phi) \mid \mathbf{C}(t,\phi) \mid$ $\mathbf{B}(a,t,\phi) \mid \mathbf{D}(a,t_1,\phi) \mid \mathbf{I}(a,t_1,\phi) \mid \mathbf{S}(a,b,t,\phi)$

Rules of Inference

$$\frac{}{\mathbf{C}(t,\mathbf{P}(a,t,\phi) \Rightarrow \mathbf{K}(a,t,\phi))}\ [R_1] \qquad \frac{}{\mathbf{C}(t,\mathbf{K}(a,t,\phi) \Rightarrow \mathbf{B}(a,t,\phi))}\ [R_2]$$

$$\frac{\mathbf{C}(t,\phi)\ t \leq t_1 \ldots t \leq l_n}{\mathbf{K}(a_1,t_1,\ldots k(a_n,t_n,\phi)\ldots)}\ [R_3] \qquad \frac{\mathbf{K}(a,t,\phi)}{\phi}\ [R_4]$$

$$\frac{}{\mathbf{C}(\mathbf{C}(t,\mathbf{K}(a,t_1,\phi_1 \Rightarrow \phi_2)) \Rightarrow \mathbf{K}(a,t_2,\phi_1) \Rightarrow \mathbf{K}(a,t_3,\phi_3))}\ [R_5]$$

$$\frac{}{\mathbf{C}(\mathbf{C}(t,\mathbf{B}(a,t_1,\phi_1 \Rightarrow \phi_2)) \Rightarrow \mathbf{B}(a,t_2,\phi_1) \Rightarrow \mathbf{B}(a,t_3,\phi_3))}\ [R_6]$$

$$\frac{}{\mathbf{C}(\mathbf{C}(t,\mathbf{C}(t_1,\phi_1 \Rightarrow \phi_2)) \Rightarrow \mathbf{C}(t_2,\phi_1) \Rightarrow \mathbf{C}(t_3,\phi_3))}\ [R_7]$$

$$\frac{}{\mathbf{C}(t,\forall x.\ \phi \Rightarrow \phi[x \mapsto t])}\ [R_8] \qquad \frac{}{\mathbf{C}(t,\phi_1 \Leftrightarrow \phi_2 \Rightarrow \neg\phi_2 \Rightarrow \neg\phi_1)}\ [R_9]$$

$$\frac{}{\mathbf{C}(t,[\phi_1 \wedge \ldots \wedge \phi_n \Rightarrow \phi] \Rightarrow [\phi_1 \Rightarrow \ldots \Rightarrow \phi_n \Rightarrow \phi])}\ [R_{10}]$$

$$\frac{\mathbf{B}(a,t,\phi_1)\ \mathbf{B}(a,t,\phi_2)}{\mathbf{B}(a,t,\phi_1 \wedge \phi_2)}\ [R_{11}]$$

$$\frac{\mathbf{S}(s,h,t,\phi)}{\mathbf{B}(h,t,\mathbf{B}(s,t,\phi))}\ [R_{12}]$$

Consider a small scenario in which there is one Ponzi schemer agent p, one investor or buyer agent b, and one investigator agent i. The goal of the Ponzi schemer is to lure investors and avoid the investigator; that of the investigator is to avoid fund collapses $S_a < 0$ by investigating a potential Ponzi schemer. The event-calculus fluent *ponzi* represents whether p is a Ponzi schemer or not, and the fluent *collapse* represents the state of the fund, i.e. whether $S_a(t) > 0$ or not. The action *investigate* is available to the investigator when the investigator believes that p is a Ponzi schemer. In terms of our formal science-of-science framework we have:

f_i : We are interested in formulae of the following forms
1. $f_1 = \{\bar{a} = S_a(t) | \bar{a} \in \mathsf{RealTerm} \wedge \bar{t} \in \mathsf{Moment}\}$
2. $f_2 = \{\bar{p} = S_p(t) | \bar{p} \in \mathsf{RealTerm} \wedge \bar{t} \in \mathsf{Moment}\}$
3. $f_3 = \{happens(action(i, investigate, \bar{t})) | \bar{t} \in \mathsf{Moment}\}$

Γ_{ponzi} : The theory of this system is given by

$$\Gamma_{ponzi} = \{\gamma_{Ponzi}, \gamma_{NI}, \gamma_{CI}\ \gamma_{susp}, \gamma_{investigate}, \gamma_{collapse}, \gamma_{p\text{-}susp}, \gamma_{p\text{-}investigate}, \gamma_{p\text{-}collapse}\} \cup \Gamma_{dyn}$$

$\mathcal{M}_{ponzi}$: m_1 and m_2 corresponding to f_1 and f_2 are implemented using primitive methods which employ procedural rewrites and call simple arithmetic functions. m_3 is implemented using a derived method in our DPL.

A simple investor who trusts the Ponzi schemer's rate-of-return claims and thus invests in the fund can be modeled as below. Here γ_{Ponzi} denotes the fact of the initial communication of the Ponzi rate to the investor.

$$\dfrac{\dfrac{\dfrac{}{\mathbf{S}(p, b, t_0, r_n(t_0) = \alpha')}\ [\gamma_{Ponzi}]}{\mathbf{B}(b, t_0, \mathbf{B}(p, t_0, r_n(t_0) = \alpha'))}\ [R_{12}] \quad \dfrac{\dfrac{}{\forall r \mathbf{B}(b, t_0, \mathbf{B}(p, t_0, r_n(t_0) = \alpha')) \rightarrow \mathbf{B}(b, t_0, r_n(t_0) = \alpha')}\ [\gamma_{NI}]}{\mathbf{B}(b, t_0, \mathbf{B}(p, t_0, r_n(t_0) = \alpha')) \rightarrow \mathbf{B}(b, t_0, r_n(t_0) = \alpha')}\ [U_{elim}]}{\mathbf{B}(b, t_0, r_n(t_0) = \alpha')}\ [MP]$$

A naïve investor would trust the Ponzi schemer to be correct about the latter's beliefs re. the nominal interest rate. We can approximate this trust by stipulating that whatever rate α' the investor believes that the Ponzi schemer believes r_n is, the investor believes the same. The investor's trust in the Ponzi schemer's beliefs re. the interest rate is represented by the axiom γ_{NI}; 'NI' abbreviates 'naïve investor.' If the investor's criteria for investing is a belief in a rate above α, with $\alpha' > \alpha$, then he invests in the scheme; this is represented by γ_{CI}; 'CI' abbreviates "criteria for investing."

$$\frac{\mathbf{B}(b, t_0, r_n(t_0) = \alpha') \quad \overline{\mathbf{B}(b, t_0, r_n(t_0) = \alpha' \rightarrow \mathit{happens}(\mathit{action}(b, \mathit{invest}), t_0))}\ [\gamma_{CI}]\ \&\ [\mathit{Arith.}]}{\mathit{happens}(\mathit{action}(b, \mathit{invest}), t_0)}\ \mathit{MP}$$

A simple investigator looking at the amount of funds claimed to be in the system gets suspicious if they are above a threshold value τ for a number of consequent time points: γ_{sup}.

$$\gamma_{susp} : \forall (t_1, \ldots, t_n) \wedge_{i=1}^{n} \mathbf{P}(i, t_i, S_a(t_i) >= \tau) \rightarrow \mathbf{B}(i, t_{n+1}, \mathit{ponzi})$$

A simple investigator also decides to investigate the moment they become suspicious without any consideration of possible negative consequences of a failed investigation: $\gamma_{investigate}$. The intention of the Ponzi schemer is to avoid a collapse of the fund, represented by $\gamma_{collapse}$, which uses the intention operator $\mathbf{I}$.

$$\gamma_{investigate} : \forall t.\ \mathbf{B}(i, t, \mathit{ponzi}) \rightarrow \mathit{happens}(\mathit{action}(i, \mathit{investigate}), t)$$
$$\gamma_{collapse} : \forall t_1, t_2.\ \mathit{prior}(t_1, t_2) \wedge \mathbf{I}(i, t_1, \neg \mathit{holds}(\mathit{collapse}, t_2))$$

While a simple Ponzi schemer may not have any beliefs about the beliefs of the investigator, a more sophisticated Ponzi schemer will ascribe the following three beliefs to an investigator. At this point, we have all that we need to simultaneously prove and simulate such results as that if the infamous Madoff had been as sophisticated as our Ponzi schemer agent p, barring a "run" on the funds Madoff controlled, exposing him would have been well nigh impossible. Space constraints preclude presenting such simulation here.

$$\gamma_{p\text{-}susp} : \forall t.\ \mathbf{B}(p, t, \gamma_{susp})$$
$$\gamma_{p\text{-}investigate} : \forall t.\ \mathbf{B}(p, t, \gamma_{investigate})$$
$$\gamma_{p\text{-}collapse} : \forall t.\ \mathbf{B}(p, t, \gamma_{collapse})$$

6 Related Work

Given both that formal logic has been vibrantly used to model human thought and action since Aristotle[16] invented and deployed his fragment of first-order logic approximately two-and-a-half millennia back, and that economics is undeniably at least partly in the business of modeling human thought and action (in connection with commerce), we find the scarcity of logicist economics to be scandalous. Be that as it may, we do say 'scarcity,' not 'absence': our pursuit of economics via formal logic isn't an entirely solitary one. What is some related work, then?

In light of his lasting reputation and impact, it's worth mentioning that much of the work of Keynes (and arguably the tenor of all that he did) is firmly in the logicist mold, as for example Keynes held probability theory to be a branch of logic [34]. In fact, Keynes specifically viewed probability theory to be about *arguments*, and our approach to inductive logic (see §8) extends our proof-centric work in deductive logic for economics, the focus of the present paper, to argument-centric work in inductive logic for the same field.[17] Yet obviously Keynes made no use of theorem-proving technology to implement logic as computation (the aforecited work on probability came well before Church, Turing, Post, and von Neumann), and no use of AI formalisms to build human-level artificial agents. It was left to Herbert Simon, nobelist in economics, to inaugurate the logicist/theorem-proving approach to simulating human

[16] His seminal work is contained in *The Organon*, available in [33].

[17] A marvelous exposition of Keynes *qua* logician, "John Maynard Keynes: a logicist with a human face," is given as a section in [35].

cognition; it was after all his LOGIC THEORIST system that, in 1956, at the famous Dartmouth conference that gave birth to AI, generated the most attention there and thereafter.[18] What about related contemporary work in computational deductive logic at least used to build and simulate such agents, in connection with economics?

Clearly the impressive work of Kaneko and a number of those he has collaborated with is relevant; we now mention some examples. In [36], we receive an elegant and extensive introduction to propositional epistemic logic in connection with game theory. One stark contrast with the calculi in $\mathscr{CC}$ that include epistemic operators and which support some of our modeling and simulation (including specifically $\mathcal{D}^y\mathcal{CEC}$), is that these calculi minimally contain as fragments full first-order logic, and maximally third-order logic, since we find that humans and human-level artificial agents frequently engage in communication, reading, and associated decision-making at the level of second- and third-order logic (where this cognition and behavior is relevant to economic behavior).[19] A second stark contrast, which by now is apparent to the reader, is that Kaneko's orientation is game theoretic; ours is not. We view game theory as a realtively small axiomatic system encompassed by various calculi in $\mathscr{CC}$, and processing that requires game theory would simply involve proof-based computation over this system. Indeed this is indicated by the chain-store case study above. In our approach to economics, while games and formalisms for them are important, of greater importance, ultimately, would be not epistemic operators and their regimentation in logic, but operators for communicating and understanding communication, which is why even the earliest members of $\mathscr{CC}$ (e.g. [7]) have machinery for communication and perception. [6] indicts the expressive power of game theory, meta-game theory, and behavioral game theory, in the light of nuclear strategy (and also provides a sustained justification for the banishment of possible-world semantics in $\mathscr{CC}$, in favor of proof-theoretic semantics, an issue referred to momentarily).

Some interesting and impressive Work on quantificational epistemic logic, in connection with economics, is in the literature curated by Kaneko; for example, [38]. This work highlights another stark contrast between our approach and prior work, because while effort in [38] is in significant measure motivated by the uncomputability of quantificational epistemic logic with a common knowledge operator, we have simply long embraced and engineered under CPU timeouts for proof search. Even standard first-order logic is after all undecidable, but that doesn't mean that proof search doesn't succeed for given modeling and simulation. In fact, while Goodstein's Theorem, now known to be an instance of the class of the undecidable theorems that Gödel abstractly defined in his famous first incompleteness theorem, has no finitary proof at present (as all proofs of it are based on transfinite ordinals), we are investigating whether an intelligent agent could nonetheless discover and prove the theorem; see e.g. progress reported in [39].[20]

Some readers may wonder about the relationship between cognitive calculi in $\mathscr{CC}$, including specifically $\mathcal{D}^y\mathcal{CEC}$, and so-called "Belief-Desire-Intention" logics, or — as they are commonly known — "BDI" logics [42]. Obviously, given our modeling above of Ponzi scheming, we find such logics to be in many regards on the right track. Yet there are major differences, and we don't have the space to provide a detailed comparison, and instead must rest content with the enumeration of a few differences from among many, to wit:

1. $\mathcal{D}^y\mathcal{CEC}$ and our other calculi make use of *proof-theoretic semantics*, rather than possible-worlds semantics; the latter is explicitly rejected. Possible-world semantics notoriously produces odd formal models when they are used for formalizing belief, knowledge, desire, and intention; for explanation and defense, see e.g. [7,6].

[18] We have found it fascinating that economists familiar with Simon's bounded-rationality approach are usually unfamiliar with his seminal applied logicist work on automated theorem proving. We unite these two Simonesque strands in [5].

[19] It is interesting to note that Kaneko himself is aware of pressure pushing toward greater expressivity; e.g., we read: "For example, extensions, *such as predicate logics*, of epistemic logics and their applications to economics are natural problems." (emphasis ours; [36], p. 56) (Usually, in fact, we use multi-sorted logic (MSL), because it has advantages at the engineering level — but we leave this issue aside. Readers wanting a nice introduction to MSL should consult [37].) It is widely recognized that human natural language involves not only quantification over relations/properties, but also ascribes properties to the variables that range over properties. In fact, economists, in their scientifiic publications, routinely invoke natural language that appears to be third-order.

[20] Along the same line, we note that a logicist analysis of the two-agent wise-man puzzle appears in *Economic Theory* [40]. We confess to never having thought of this puzzle, a longstanding staple in logicist AI (e.g. see [16]), as relevant to economics. Bringsjord, with Arkoudas, provided to our knowledge the first formal, machine solution to the puzzle for *any* (natural) number of agents: [41].

Our use of proof-theoretic semantics means that, in general, model-based reasoning for epistemic operators [43] is also not used by any cognitive calculus in $\mathscr{CC}$, and therefore not by any dialect of $\mathcal{CEC}$ or $\mathcal{D}^y\mathcal{CEC}$.[21]

2. Natural deduction, a revolution that burst on the formal-logic scene in 1934 [48,49] is used; this form of deduction can faithfully capture many aspects of reasoning used by human beings [17]. This is not the case for such things as resolution, which is based on inference schemas never instantiated, e.g., in the proofs and theorems that anchor the formal sciences (e.g., game theorists never give proofs based in resolution, but rather in natural deduction). Whereas $\mathcal{D}^y\mathcal{CEC}$ inference parallels normative human reasoning by providing natural justifications via the proofs involved in inference, this is not always the case in BDI logics. In addition, natural deduction based in hypergraphs, unique to Bringsjord's Slate system, is never seen in BDI logics (see footnote 14 for a sample proof). And, we have not seen *methods* (recall our use of them in the chain-story study above) used in BDI systems.
3. Uncertainty is handled in $\mathscr{CC}$not only via axiomatized probability calculi given in [50] (available via Gödel numbering in the object language of cognitive calculi not used in the present paper), but by a 9-valued logic generally in harmony with, but an aggressive extension of, Pollock's defeasible logic [51,52]. Each of the nine values is a *strength factor* [28].
4. Operators for obligation, perception, communication, and other intensional operators/activities are included in $\mathcal{D}^e\mathcal{CEC}$ and other deontic cognitive calculi in $\mathscr{CC}$; in the case of communication, the relevant operators are associated with built-in semantic parsing, and the λ-calculus is subsumed and employed. In stark contrast, BDI logics don't for instance subsume deontic logics (which traditionally formalize obligation; see [53] for robust modeling and simulation using deontic operators working in conjunction with epistemic ones), and don't have operators corresponding to full natural-language understanding (English to formulae in cognitive calculi) and full generation (formulae in cognitive calculi to English). (Of course, we earlier introduced and exploited a parallel to the speech acts of [32].)
5. Finally, diagrammatic representation is in and crucial to $\mathscr{CC}$, whereas BDI logics are all provably exclusively linguistic/sentential in nature, since all formulae in such logics are formed from alphabets of only symbols or characters.. A logic — Vivid — allowing both standard linguistic formulae and diagrammatic representations is presented and proved sound in [54]. Vivid continues to heavily influence $\mathscr{CC}$, and is being used for our work in axiomatic physics, where for example special relativity is reduced to formal logic [55].

7 Objections; Rebuttals

We consider and rebut some inevitable objections to logicist agent-based economics as we define it:

7.1 Objection #1: Complexity

We imagine some will say: "It is very hard to see how your approach can be applied to more complex, real-world problems. You do not speak to issues of computational complexity at all, and there is also the worry that, by definition, a 'complex' system cannot be totally subsumed by (= reduced to) a formal model."

We agree that it is indeed hard to see how our approach can scale to the level of yielding real world solutions. Of course that which is hard to see is not invisible, and what can be seen is also undeniably perspective- and person-relative. We believe we already see the possibility of deploying an AI system into the financial markets that continuously scans for the potential satisfaction of a $\mathscr{CC}$-based definition $\mathcal{P}$ of Ponzi-scheme behavior, extrapolated slightly from the work shown above. The only way to *demonstrate* such feasibility is either to provide some kind of inductive proof, or some kind of naïve empirical form of induction. In the latter case, we, or some other group, would implement a suitable system and test it empirically — in larger and larger markets. The door is open for this, it would seem. Parallel comments can be easily imagined by the reader for predictions about whether some entrepreneurs considering going up against a real-life CS behemoth will encounter a fight or note, where the number of agents/companies in question is very large. It is true that we don't speak of computational complexity. For the most part, doing so would be otiose, since theoremhood in even first-order logic is Turing-undecidable (semi-decidable); in the richer calculi in $\mathscr{CC}$, theoremhood is fully Turing-undecidable. Computational complexity gives us a sub-hierarchy of the small part of the Arithmetic Hierarchy that contains Turing-solvable functions. Finally, we are ignorant of any theorem to the effect that there is some environment

[21] Hard-working readers unfamiliar with proof-theoretic semantics are encouraged to consult a body of work that we find makes for a nice introduction: [44,45,46,47].

or system such that there is no formal model to which it can be reduced.[22] It may be worth remembering that even Gödelian incompleteness is surmounted by simple infinitary logics.

7.2 Objection #2: Benefits

"You have failed to clearly demonstrate the benefits of your approach in economics. For instance, in the chain-store case, what more is offered above and beyond computational game theory?"

Our rebuttal to prior objection does serves as a partial reply here. In addition, we have explained that in our approach, proof-based as it is, processing and outputs are all accompanied by checkable, verifiable proofs. Game theory, axiomatized, is subsumed by our calculi; but game theory, even in its recent "behavioral" guise (e.g. [27]), omits those aspects of human cognition that are central to, indeed constitutive of, economic phenomena. Natural-language communication is a prime example. Another is planning and re-planning, traditionally a logicist enterprise in AI ([2] provides an overview), but completely absent in game theory. We could continue.

7.3 Objection #3: Irrationality

"There is overwhelming evidence that humans are frequently, if not fundamentally, irrational. Piaget claimed that neurobiologically normal humans would in the course of natural development acquire an ability to for example reason in keeping with first-order logic [56], but this was refuted originally by Wason [57], and evidence has continued to mount all the way through Simon and his concept of bounded rationality to nobelist (in economics) Kahneman. This means that logic has very limited applicability."

Formally speaking, logic isn't in the least wed to rationality. It is hard for logicians with command over the full gamut of deductive and inductive logics to fathom how the notion that logic and rationality are inseparable continues to live on. Rips [58] for instance gives logics in which theorems that irrational humans fail to grasp (e.g., that from $\neg(\phi \rightarrow \psi)$ it deductively follows in standard, elementary deductive logic that ϕ) are unprovable. We have ourselves provided rather elaborate logicist modeling and simulation (of a certain class of auctions) that is nonetheless explicitly "Simonesque" [5]. In addition, paraconsistent logics formalize the ultimate in irrationality: toleration/managment of explicit contradiction. In inductive logic, the compatibility of logic and irrationality is easy enough to demonstrate, but we haven't the space to elaborate.

8 Some Next Steps

While $\mathcal{CEC}$ and $\mathcal{D}^y\mathcal{CEC}$ may be commendably expressive along certain cognitive dimensions, there are clearly deficiencies along others. For example, nothing in them accommodates uncertainty/probability. Yet even in chain-store scenarios, agents believe in a manner well short of certainty: CS may for instance believe only that it's *likely* that if it fights entrant E_i in plain view of E_j, E_j will decide to stay out. We are currently refining implemented dialects of our calculi that provide the full machinery of attractive inductive logics (close to how Fitelson [59] defines such logics, which e.g. Keynes too would find acceptable), including built-in computational axiomatizations of Kolmogorovian probability at both the propositional and first-order level [again, as expressed in [50]], and including strength-based approaches as well [again, as described in [28]].[23] Another important next step is the publishing of theorems (which are not difficult) showing that our paradigm subsumes both the micro-simulation approach to economics [61], and "thin" agent-based approaches [62]. In both cases, only the first-order extensional fragments of cognitive calculi in $\mathscr{CC}$ are needed for such theorems, and corresponding modeling and simulation.

References

1. Bringsjord, S.: Declarative/Logic-Based Cognitive Modeling. In Sun, R., ed.: The Handbook of Computational Psychology. Cambridge University Press, Cambridge, UK (2008) 127–169

[22] We do acknowledge that e.g. if one restricts oneself to a particular fixed formal formal model (say differential equations) for some particular physical environment or system, reduction is impossible.

[23] We reduce both decriptive and inferential statistics to inductive logic. For foundational explanation of this integrative trajectory, see [60].

2. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River, NJ (2009) Third edition.
3. Arkoudas, K., Bringsjord, S.: Computers, Justification, and Mathematical Knowledge. Minds and Machines **17** (2007) 185–202
4. Almeida, J., Frade, M., Pinto, J., de Sousa, S.: Rigorous Software Development: An Introduction to Program Verification. Spinger, New York, NY (2011)
5. Johnson, J., Govindarajulu, N., Bringsjord, S.: A Three-Pronged Simonesque Approach to Modeling and Simulation in Deviant 'Bi-Pay' Auctions, and Beyond. Mind and Society **13** (2014) 59–82
6. Bringsjord, S., Govindarajulu, N., Ellis, S., McCarty, E., Licato, J.: Nuclear Deterrence and the Logic of Deliberative Mindreading. Cognitive Systems Research **28** (2014) 20–43
7. Arkoudas, K., Bringsjord, S.: Propositional Attitudes and Causation. International Journal of Software and Informatics **3** (2009) 47–65
8. Bringsjord, S., Govindarajulu, N.S.: Toward a Modern Geography of Minds, Machines, and Math. In M¨ller, V.C., ed.: Philosophy and Theory of Artificial Intelligence. Volume 5 of Studies in Applied Philosophy, Epistemology and Rational Ethics. Springer, New York, NY (2013) 151–165
9. Jain, S., Osherson, D., Royer, J., Sharma, A.: Systems That Learn: An Introduction to Learning Theory, Second Edition. MIT Press, Cambridge, MA (1999)
10. Bringsjord, S., Govindarajulu, N.S., Eberbach, E., Yang, Y.: Perhaps the Rigorous Modeling of Economic Phenomena Requires Hypercomputation. International Journal of Unconventional Computing **8** (2012) 3–32
11. Suppes, P.: Representation and Invariance of Scientific Structures. CSLI, Stanford, CA (2001) This is book 130 from CSLI. Subsequent versions of this book are distributed by the University of Chicago Press.
12. Tarski, A.: Introduction to Logic and to the Methodology of Deductive Sciences. Dover Publications, New York, NY (1995)
13. Ebbinghaus, H.D., Flum, J., Thomas, W.: Mathematical Logic (second edition). Springer-Verlag, New York, NY (1994)
14. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River, NJ (2002)
15. Mueller, E.: Commonsense Reasoning. Morgan Kaufmann, San Francisco, CA (2006)
16. Genesereth, M., Nilsson, N.: Logical Foundations of Artificial Intelligence. Morgan Kaufmann, Los Altos, CA (1987)
17. Arkoudas, K.: Denotational Proof Languages. PhD thesis, MIT (2000)
18. Muggleton, S.: Inductive logic programming. Morgan Kaufmann (1992)
19. Halpern, J.: An Analysis of First-Order Logics of Probability. Artificial Intelligence **46** (1990) 311–350
20. Selten, R.: The Chain Store Paradox. Theory and Decision **9** (1978) 127–159
21. Nozick, R.: Newcomb's problem and two principles of choice. In Rescher, N., ed.: Essays in Honor of Carl G. Hempel. Humanities Press, Highlands, NJ (1970) 114–146
22. Aumann, R.J.: Backward induction and common knowledge of rationality. Games and Economic Behavior **8** (1995) 6–19
23. Balkenborg, D., Winter, E.: A necessary and sufficient epistemic condition for playing backward induction. Journal of Mathematical Economics **27** (1997) 325–345
24. Kreps, D., Wilson, R.: Reputation and imperfect information. Journal of Economic Theory **27** (1982) 253–279
25. Tracy, W.: Paradox lost: The evolution of strategies in selten's chain store game. In: Three Essays on Firm Learning. (2008) This is a dissertation at UCLA in the Anderson School of Management.
26. Osborne, M., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge, MA (1994)
27. Camerer, C.: Behavioral Game Theory: Experiments in Strategic Interaction. Princeton University Press, Princeton, NJ (2003)
28. Bringsjord, S., Taylor, J., Shilliday, A., Clark, M., Arkoudas, K.: Slate: An Argument-Centered Intelligent Assistant to Human Reasoners. In Grasso, F., Green, N., Kibble, R., Reed, C., eds.: Proceedings of the 8th International Workshop on Computational Models of Natural Argument (CMNA 8), Patras, Greece, University of Patras (2008) 1–10
29. Aumann, R.J.: Backward induction and common knowledge of rationality. Games and Economic Behavior **8** (1995) 6–19
30. Stickel, M., Waldinger, R., Lowry, M., Pressburger, T., Underwood, I.: Deductive Composition of Astronomical Software From Subroutine Libraries. In: Proceedings of the Twelfth International Conference on Automated Deduction (CADE–12), Nancy, France (1994) 341–355 SNARK can be obtained at the url provided here.
31. Artzrouni, M.: The mathematics of ponzi schemes. Mathematical Social Sciences **58** (2009) 190–201
32. Wooldridge, M.: An introduction to multiagent systems. Wiley (2002)
33. McKeon, R., ed.: The Basic Works of Aristotle. Random House, New York, NY (1941)
34. Keynes, J.M.: A Treatise on Probability. Macmillan, London, UK (1921)

35. Galavotti, M.C.: The Modern Epistemic Interpretations of Probability: Logicism and Subjectivism. In Gabbay, D., Hartmann, S., Woods, J., eds.: Handbook of the History of Logic; Volume 10: Inductive Logic. Elsevier, Amsterdam, The Netherlands (2011) 153–203
36. Kaneko, M.: Epistemic Logics and Their Game Theoretic Applications: Introduction. Economic Theory **19** (2002) 7–62
37. Manzano, M.: Extensions of First Order Logic. Cambridge University Press, Cambridge, UK (1996)
38. Sturm, H., Wolter, F., Zakharyaschev, M.: Common Knowledge and Quantification. Economic Theory **19** (2002) 157–186
39. Govindarajulu, N., Licato, J., Bringsjord, S.: Small Steps Toward Hypercomputation via Infinitary Machine Proof Verification and Proof Generation. In Giancarlo, M., Dennuzio, A., Manzoni, L., Porreca, A., eds.: Unconventional Computation and Natural Computation; Lecture Notes in Computer Science; Volume 7956, Berlin, Germany, Springer-Verlag (2013) 102–112
40. Yasugi, M., Oda, S.: A Note on the Wise Girls Puzzle. Economic Theory **19** (2002) 145–156
41. Arkoudas, K., Bringsjord, S.: Metareasoning for Multi-agent Epistemic Logics. In: Fifth International Conference on Computational Logic In Multi-Agent Systems (CLIMA 2004). Volume 3487 of Lecture Notes in Artificial Intelligence (LNAI). Springer-Verlag, New York (2005) 111–125
42. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: Proceedings of Knowledge Representation and Reasoning (KR&R-91). (1999) 473–484
43. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning About Knowledge. MIT Press, Cambridge, MA (2004)
44. Gentzen, G.: Investigations into Logical Deduction. In Szabo, M.E., ed.: The Collected Papers of Gerhard Gentzen. North-Holland, Amsterday, The Netherlands (1935) 68–131 This is an English version of the well-known 1935 German version.
45. Prawitz, D.: The Philosophical Position of Proof Theory. In Olson, R.E., Paul, A.M., eds.: Contemporary Philosophy in Scandinavia. Johns Hopkins Press, Baltimore, MD (1972) 123–134
46. Kreisel, G.: A Survey of Proof Theory II. In Renstad, J.E., ed.: Proceedings of the Second Scandinavian Logic Symposium. North-Holland, Amsterdam, The Netherlands (1971) 109–170
47. Francez, N., Dyckhoff, R.: Proof-theoretic Semantics for a Natural Language Fragment. Linguistics and Philosophy **33** (2010) 447–477
48. Gentzen, G.: Untersuchungen über das logische Schlieben I. Mathematische Zeitschrift **39** (1935) 176–210
49. Jaśkowski, S.: On the Rules of Suppositions in Formal Logic. Studia Logica **1** (1934) 5–32
50. Adams, E.: A Primer of Probability Logic. CSLI, Stanford, CA (1998)
51. Pollock, J.L.: How to Reason Defeasibly. Artificial Intelligence **57** (1992) 1–42
52. Pollock, J.: Defasible Reasoning with Variable Degrees of Justification. Artificial Intelligence **133** (2001) 233–282
53. Bringsjord, S., Govindarajulu, N., Thero, D., Si, M.: Akratic Robots and the Computational Logic Thereof. In: Proceedings of *ETHICS* • 2014 (2014 IEEE Symposium on Ethics in Engineering, Science, and Technology), Chicago, IL (2014) 22–29 IEEE Catalog Number: CFP14ETI-POD. Papers from the *Proceedings* can be downloaded from IEEE at `http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6883275`.
54. Arkoudas, K., Bringsjord, S.: Vivid: An AI Framework for Heterogeneous Problem Solving. Artificial Intelligence **173** (2009) 1367–1405 The url http://kryten.mm.rpi.edu/vivid/vivid.pdf provides a preprint of the penultimate draft only. If for some reason it is not working, please contact either author directly by email.
55. Govindarajalulu, N.S., Bringsjord, S., Taylor, J.: Proof Verification and Proof Discovery for Relativity. Synthese (2014) 1–18
56. Inhelder, B., Piaget, J.: The Growth of Logical Thinking from Childhood to Adolescence. Basic Books, New York, NY (1958)
57. Wason, P.: Reasoning. In: New Horizons in Psychology. Penguin, Hammondsworth, UK (1966)
58. Rips, L.: The Psychology of Proof. MIT Press, Cambridge, MA (1994)
59. Fitelson, B.: Inductive logic. In Pfeifer, J., Sarkar, S., eds.: Philosophy of Science: An Encyclopedia. Routledge, London, UK (2005) 384–394
60. Romeijn, J.W.: Inductive logic and statistics. In Gabbay, D., Thagard, P., Woods, J., eds.: Handbook of the History of Logic. Elsevier, Amsterdam, The Netherlands (2009) 625–650 This is in Volume 10: Inductive Logic in the series.
61. Orcutt, G., Greenberger, M., Korbel, J., Korbel, J., Rivlin, A.: Microanalysis of Socioeconomic Systems: A Simulation Study. Harper and Brothers, New York, NY (1961)
62. Epstein, J., Axtell, R.: Growing Artificial Societies. Bradford Books, Cambridge, MA (1996)