Betting on the latent space

(working title; year 1 annual report)

Name: Abhimanyu Pallavi Sudhir (u2251609)
Supervisor: Long Tran-Thanh
Course: PhD Computer science, 1 Dec 2022–26, University of Warwick

1 Introduction

Prediction markets can be used to elicit and aggregate agents' information about some event [MI1–MI3]; for example, the market price for the contract "Pays $1 if a Republican wins the 2024 US Presidential Election" can be interpreted as the probability assigned by the market to the event that a Republican wins the 2024 US Presidential Election (we will make this notion precise in subsequent sections).

On the surface, prediction markets seem to solve basically every problem in your life: any question, any difference of opinion, even any scientific question, is optimally addressed by simply setting up a prediction market for it, yielding the single best probability estimate as an answer from all information present in the world.

Closer inspection reveals a fundamental limitation of prediction markets: they can only elicit probabilities for sentences that are either *verifiable* (if true, then will be revealed true – e.g. "there exists a white swan") or *falsifiable* (if true, then will be revealed false – e.g. "all swans are white"). However, intelligent agents generally hold beliefs about a much wider class of sentences: sentences that only hold meaning in an agent's mental model, that could be described as "sentences that are neither verifiable nor falsifiable (non-VF)", "beliefs in the latent space" or even "subjective beliefs".

One type of such a class of sentences is sentences expressible in some logic, e.g. First-Order Logic (FOL), as in Fig 1. But even this does not capture the full scope of sentences in an agent's latent space: e.g. claims about events in the past, claims about a tree falling in the forest with no one to hear it, claims about quarks, claims about hypotheticals that will certainly never be carried out, vaguely-specified claims.

One approach to dealing with such sentences, which could be said to be the philosopher's approach, is to reject non-VF sentences as meaningful at all. However, we may defend such sentences on the basis of their practical value: take the sentence "Bob is guilty". This sentence in itself is non-VF; however, it is *correlated* with other, directly empirical or practically valuable sentences such as "If we release Bob from prison, he will commit more crimes" and "we will find blood in Bob's house": intuitively, we expect that agents, seeing a high price for "Bob is guilty", will also bid up "we will find blood in Bob's house", or bid against releasing Bob on a decision market. Thus "Bob is guilty" can be seen as a latent space variable that captures correlations between different variables. This is expounded on in [L1].

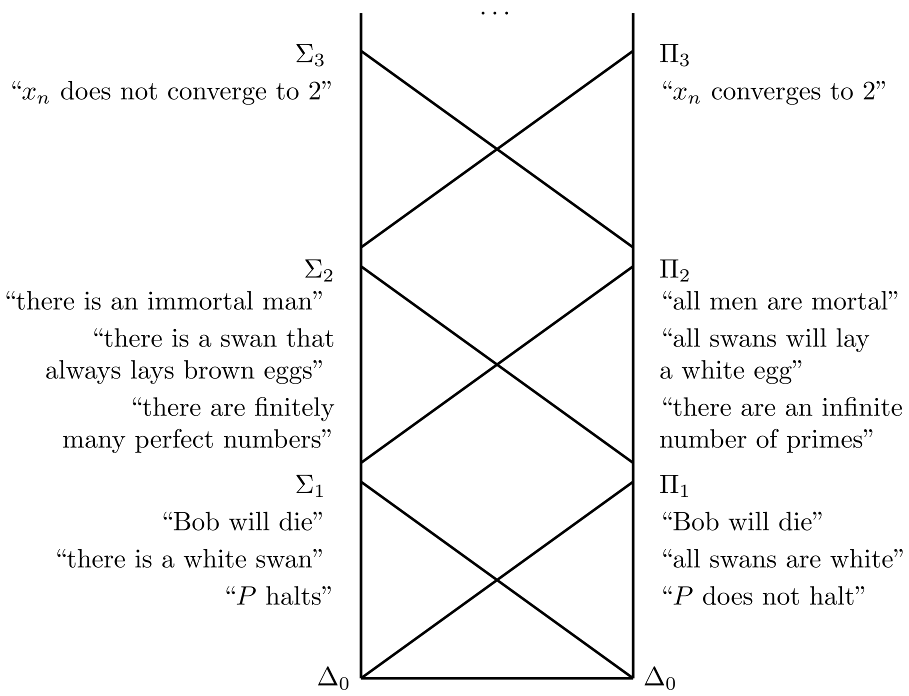

Figure 1: The "arithmetical hierarchy" of FOL sentences. A Σ_{n+1} sentence is of the form $\exists x, p(x)$ where p is Π_n; a Π_{n+1} sentence is of the form $\forall x, p(x)$ where p is Σ_n. In logic, the lowest level of the arithmetical hierarchy $\Delta_0 = \Sigma_0 = \Pi_0$ are sentences of the form $f(n) = 0$ where f is a primitive recursive function, but this formalism may be extended to empirical truths that will be revealed at a fixed time

Research objective. Broadly speaking, I seek to *create market-based mechanisms for creating and betting on latent space variables.* More precisely, this can be expressed in terms of two slightly differing research questions:

1. (As a toy model) How can we develop markets that meaningfully price sentences expressible in FOL (or more generally any hyperarithmetical logic)?

2. How can we develop markets that decide (with the objective of maximizing some utility function) what kind of latent space to build, and then elicit bets on questions in the latent space?

3. How can we develop prediction markets that incentivize a collection of agents to develop "mutually interpretable" latent spaces, i.e. such that they can cheaply and honestly place bets on sentences framed in each other's latent spaces with good calibration and scores.

Work so far. I've addressed RQ 1 in my draft paper [O1] and produced a framework for betting on FOL sentences; I'll elaborate on the precise framework in subsequent sections.

Applications. Besides the obvious utility of constructing prediction markets for arbitrary subjective sentences, the primary application of my work is to *AI alignment.* More precisely:

- The desiderata outlined above for my market-based mechanism – forming a latent space that captures patterns in observed data – are essentially identical to the function of an intelligent agent. My hope therefore is that my research could lead to an alternate, market-based framework for AI agents, and there are intuitive reasons to expect such a framework to be immune to many forms of misalignment.

- Eliciting bets on latent space knowledge, if robust frameworks to incentivize this are developed, have been identified as an approach to AI interpretability [L2].

2 Literature survey

2.1 Prediction markets

We will describe the general setting of a basic prediction market, specifically under the Logarithmic Market Scoring mechanism of [MI2, MI3] as that framework is most natural for simultaneously addressing *which questions should prediction markets be created for*, etc.

Definition 2.1 (Finite process)**.** A "finite process" X is the tuple of a resolution time $X.t \in \mathbb{N}$, and a true value $X.x \in \{\bot, \top\}$ (which may be sampled from some probability distribution at time t).

We concern ourselves with some particular set FinProc of finite processes, e.g. statements of the form $f(n) = 0$ where f is a primitive recursive function over the natural numbers and the resolution time is given by some enumeration of the primitive recursive functions. In particular FinProc should contain elements $\top$ and $\bot$ which have resolution time $t = 0$ and true value $v = 1$ and 0 respectively. Furthermore we denote the type of finite-supported maps $\text{FinProc} \times \{\bot, \top\} \to_{\circ} \mathbb{Q}$ as PF (for "portfolio").

Definition 2.2 (Agent)**.** An "agent" α is a computable function $\hat{\alpha} : \mathbb{N} \to \text{PF}$ paired with a "starting endowment" $\alpha^{\text{b}} \in \mathbb{Q}$. [Here, $\hat{\alpha}(t)$ should be interpreted as the order submitted by the agent, i.e. which should get added to its portfolio if all goes well]

Again, we concern ourselves with some particular class $\mathcal{A}$ of agents and specifically a surjective "enumerator" of agents $\mu^{\text{bh}} : \mathbb{N} \to \mathcal{A}$ such that $\sum_{t\in\mathbb{N}} \mu^{\text{bh}}(t)^{\text{b}} < \infty$ that allows us to have a potentially infinite number of traders that are gradually added to trade in the market.

Definition 2.3 (Classic prediction market)**.** A prediction market for a finite process X is defined by a "costing function" $P : \text{PF} \to \mathbb{R}$ and the following procedures for updating each agent's inventory $\alpha_{\$} : \text{PF}$ (note that in particular, $\alpha_{\$}(\top, \top)$ is interpreted as α's cash reserves) and market prices $\pi : \text{FinProc} \times \{\bot, \top\} \to [0, 1]$:

```
function REWARDER
    for $t \in \mathbb{N}$ do $\mu^{\text{bh}}(t)_{\$}(\top, \top) \leftarrow \mu^{\text{bh}}(t)^{\text{b}}$
        for $\alpha \in \{\mu^{\text{bh}}(0), \dots \mu^{\text{bh}}(t)\}$ do
            for $X \in \operatorname{supp} \alpha_{\$}$ do              ▷ for all stocks $\alpha$ has a stake in
                if $t \geq X.t$ then                                ▷ if they're resolved
                    $\alpha_{\$}(\top, \top) \leftarrow \alpha_{\$}(X, X.x)$        ▷ convert them into cash
                    $\alpha_{\$}(X, \top) \leftarrow 0$
                    $\alpha_{\$}(X, \bot) \leftarrow 0$
function MARKET
    $\pi_{\$} \leftarrow \mathbf{0}$                    ▷ initialize total count of stocks in circulation
    for $t \in \mathbb{N}$ do
        for $\alpha \in \{\mu^{\text{bh}}(0), \dots \mu^{\text{bh}}(t)\}$ do
            $\Delta\alpha_{\$} \leftarrow \alpha(t)$                                  ▷ input trade order
            $\Delta\alpha_{\$}(\top, \top) \leftarrow P(\alpha_{\$}) - P(\alpha_{\$} + \Delta\alpha_{\$})$     ▷ calculate order cost
            if $\alpha_{\$}(\top, \top) + \Delta\alpha_{\$}(\top, \top) \geq 0$ then            ▷ Check if in budget
                $\alpha_{\$} \leftarrow \alpha_{\$} + \Delta\alpha_{\$}$
                $\pi_{\$} \leftarrow \pi_{\$} + \Delta\alpha_{\$}$
            $\pi \leftarrow \nabla P(\pi_{\$})$
```

The above definition is for some general "costing function" P whose gradient is the instantaneous price: in general, any costing function corresponds to a *scoring rule*: if you push the price of some asset from π to π' with your trades, and that asset resolves to $\top$, your profit can be described as a score $s(\pi') - s(\pi)$ you receive for your prediction. More precisely, and slightly generalizing the statement in [MI2, MI3]:

Definition 2.4 (Corresponding scoring rule)**.** A necessary condition for an acceptable costing function is that $\nabla P(\mu'_\$) = \nabla P(\mu_\$)$ only when $\mu'_\$ - \mu_\$$ amounts to a cash term (i.e. an equal distribution over all mutually exclusive stocks), and that in this case $P(\mu'_\$) - P(\mu_\$) = \mu'_\$ - \mu_\$$. In that case, the costing function leads to a scoring rule for report π given by $s(\pi) = \theta(\pi) - P \circ \theta(\pi)$ where θ is any right-inverse of ∇P, i.e. $\theta(\pi)$ is a portfolio that leads to a market price of π.

With this, all the standard textbook knowledge about scoring rules can be applied (indeed this was the central insight of [MI2, MI3]). For instance, we would like a costing function such that its corresponding scoring rule is *proper*, i.e. if $\mathbf{E}[s(\mathbf{r})]$ is maximized under a probability distribution $\pi : \text{FinProc} \to [0,1]$ when $\mathbf{r} = \pi$.

The choice of this costing function determines the mechanism, and can generally be interpreted as expressing *how the market-maker demands information.* For instance, under *logarithmic market scoring*, the market maker is essentially paying a linear cost for information on some finite process, i.e. some price per bit of information on X. More precisely:

Definition 2.5 (Logarithmic Market Scoring (LMSR))**.** Logarithmic market scoring [MI2] is defined by the costing function:

$$P(\alpha_\$) = \sum_{X \in \operatorname{supp} \alpha_\$} \lambda_X \log \sum_{x \in \{\bot, \top\}} \exp\left(\alpha_\$(X, x)/\lambda_X\right)$$

Where λ_X the "market-maker subsidy" for the market on X (if the finite processes can be computably enumerated, λ_X can be pre-specified so it sums to a finite value over all X). In fact it represents the price per bit of information on the value of X. One can check that prices are then given by:

$$\nabla P(\alpha_\$, X, x) = \frac{\exp\left(\alpha_\$(X, x)/\lambda_X\right)}{\sum_{x \in \{\bot, \top\}} \exp\left(\alpha_\$(X, x)/\lambda_X\right)}$$

One may check that the resulting scoring rule (the total market payout made in each possible outcome, for moving prices to $\mathbf{r}$) is $\mathbf{s}(\mathbf{r}) = [\lambda_X \log \mathbf{r}(X, x)]_{X \in \text{finset FinProc}, x \in \{\bot, \top\}}$, the expectation of which, under belief π, is:

$$s(\pi, \mathbf{r}) = \sum_X \sum_{x \in \{\bot, \top\}} \lambda_X \pi(X, x) \log \mathbf{r}(X, x) = -\sum_X \lambda_X H(\pi(X), \mathbf{r}(X))$$

Thus the expected profit from making a report $\mathbf{r}'$ when current market prices are $\mathbf{r}$:

$$\Delta s(\pi, \mathbf{r}, \mathbf{r}') = s(\pi, \mathbf{r}') - s(\pi, \mathbf{r}) = \sum_X \lambda_X \left[H(\pi(X), \mathbf{r}(X)) - H(\pi^*(X), \mathbf{r}'(X))\right]$$

Where H is cross-entropy. We would *like* to say that $\Delta s(\pi, \mathbf{r}, \mathbf{r}')$ is maximized when $\mathbf{r}' = \pi$, but the report $\mathbf{r}'$ has finite support while π is not, so there is no "best possible report" to make. But we can see that the profit is decreasing in the cross-entropy $H(\pi(X), \mathbf{r}'(X))$, i.e. the closer your report to your true belief, the better.

2.2 Program markets

We will describe here, in brief, two existing works in the literature describing "program markets", i.e. markets whose participants are taken to be programs rather than some utility-maximizing agents. Treating the market as a whole as an aggregate agent then quite nicely captures a notion of bounded rationality, because a market can be understood as being optimally rational *as constrained by the algorithmic information (i.e. programs) available to it.*

The first work we will describe is [PM1], a framework for logical uncertainty, to which my work [O1] can be seen as a competing framework. The work defines a prediction market for all logical sentences which pays off whenever a sentence is proven by some formal theorem prover (which exists for any "computably enumerable theoory", including first-order arithmetic etc.). However, as stated, this would simply be a prediction market for the *provability* of a sentence (in some particular formal theory); the actual Garrabrant induction framework instead adopts a notion of "propositionally consistent worlds" that allows even unprovable (and provably unprovable) sentences to have non-zero probabilities and such that basic probabilistic laws like $\mathbf{Pr}(P) + \mathbf{Pr}(\neg P) = 1$ are followed.

The idea is as follows: if $P \lor Q$ is proven, then an agent that holds a stock each in sentences P and Q must necessarily be regarded as having *at least* \$1 between these assets. Similarly if $\neg P \lor \neg Q$ is proven, then that agent has *at most* \$1 between these assets. We call these different logical possibilities "worlds" or rather "worlds that are propositionally consistent with the output of the theorem prover so far" (PC worlds for short). In particular when evaluating whether a trade is within budget (to be accepted by the market-maker), we demand that it is within budget in all PC worlds. More precisely:

Definition 2.6 (Worlds and valuations)**.** Let $\alpha_\$: \text{Prop} \to_\circ \mathbb{Q}$ be a finite-supported map, and let $w : \text{Prop} \to \{\bot, \top\}$ be a *world*, i.e. a truth-assignment, so the dot product $w \cdot \alpha_\$$ represents the valuation of $\alpha_\$$ according to w.

Definition 2.7 (PC worlds)**.** A world is said to be propositionally consistent (PC) if for all $P \in \text{Prop}$, $w(P)$ is determined by Boolean algebra from the prime sentences in Prop, i.e. $w(P \land Q) = w(P) \land w(Q)$, $w(P \lor Q) = w(P) \lor w(Q)$, etc. Furthermore, let Θ_t be the subset of Prop proven by time t: then a world is said to be PC with Θ_t if (1) it is PC and (2) $w(P) = 1$ for all $P \in \Theta_t$. We denote the set of worlds PC with Θ_t as $\mathsf{PC}(\Theta_t)$; in particular $\mathsf{PC} := \mathsf{PC}(\{\})$. For any $\alpha_\$$, denote its set of plausible valuations as $\mathsf{PC}(\Theta_t, \alpha_\$) := \{w \cdot \alpha_\$ \mid w \in \mathsf{PC}(\Theta_t)\}$.

Note that $\mathsf{PC}(\Theta_t, \alpha_\$)$ is computable, since you only have to check propositional consistency for the sentences actually supported by $\alpha_\$$.

Another quirk of Garrabrant induction is that agents have type $\alpha : \mathbb{N} \to (\text{Prop} \to [0,1]) \to (\text{Prop} \to \mathbb{Q})$ i.e. they output "joint" demand schedules allowing for cross-elasticity of demand rather than leaving this to be figured out by the market dynamics. Naturally, calculating equilibrium between such demand schedules is non-trivial and requires Brouwer's fixed point theorem, as well as constraints on α (specifically that $\alpha(t)$ is continuous in price and comprised only of some particularly simple expressions depending only on some external information like price history). The framework actually implements a rational approximation of the equilibrium computed via a brute-force Farey enumeration of all rational numbers.

Definition 2.8 (Garrabrant induction)**.** Fix a language Prop, a theorem enumerator $\Theta : \mathbb{N} \to \operatorname{finset} \text{Prop}$ (obeying in particular $s \leq t \implies \Theta_s \subseteq \Theta_t$) and enumerator of agents $\mu^{\text{bh}} : \mathbb{N} \to \mathcal{A} \times \mathbb{Q}$ such that μ_1^{bh} is bijective and $\sum_t \mu_2^{\text{bh}}(t) < \infty$ (where $\mu^{\text{bh}} = (\mu_1^{\text{bh}}, \mu_2^{\text{bh}})$ and $\mathcal{A}$ is the type specified earlier $\mathbb{N} \to (\text{Prop} \to [0,1]) \to (\text{Prop} \to \mathbb{Q})$). Then the Garrabrant induction algorithm is given by the following mutual recursion:

- an "aggregate trader"

  $$\mu(t,\pi) := \sum_{\alpha \in \{\mu^{\text{bh}}(1) \ldots \mu^{\text{bh}}(t)\}} \mathbf{I}\left[\min \mathsf{PC}(\Theta_t, \alpha_\$(t,\pi) + \alpha(t,\pi)) \geq 0\right] \alpha(t,\pi)$$

  where $\mathbf{I}\left[\right]$ denotes an indicator function

- an "equilibrium price" $\pi(t)$, which approximates a zero of $\mu(t)$ i.e. so that $\mu(t,\pi(t)) \approx 0$ (in particular the error should be $\leq 1/2^t$)

- an inventory account $\alpha_\$(t, P)$, as computed by the following algorithm:

```
function $\alpha_\$(\tau)$
    for $t \leq \tau$ do
        $\mu_1^{\text{bh}}(t)_\$(\top) \leftarrow \mu^{\text{bh}}(t)_2$
        for $X \in \operatorname{supp} \alpha_\$$ do                         ▷ resolve proven sentences
            if $X \in \Theta_t$ then
                $\alpha_\$(\top) \leftarrow \alpha_\$(\top) + \alpha_\$(X)$
                $\alpha_\$(X) \leftarrow 0$
        for $X \in \operatorname{supp} \alpha$ do
            if $\min \mathsf{PC}(\Theta_t, \alpha_\$(t,\pi) + \alpha(t,\pi)) \geq 0$ then
                $\alpha_\$ \leftarrow \alpha_\$ + \alpha(t)$                 ▷ add trade to inventory
    return $\alpha_\$$
```

[A slight difference in the formulation we have presented is that we put the onus on the individual agents to calculate and include the cash payment within their orders, and the aggregate agent just zeroes their trade if they submit an invalid trade; this makes no difference to the whole algorithm.]

Theorem 2.1 (Inexploitability)**.** *With definitions as in Def 2.8,* no trader $\alpha \in \mathcal{A}$ can exploit the price sequence π, *i.e.* $\mathsf{PC}(\Theta_t, \alpha_{\$}(t))$ *remains bounded from above.*

Proof sketch. If any trader could exploit the price sequence, so could the aggregate trader μ (as whichever trader exploits the market would also grow its proportion of μ's trades to $+\infty$, and no trader can run μ's finances to $-\infty$ since it would just go bankrupt). However, μ cannot exploit π as it is set to be (arbitrary close to) an equilibrium for μ. □

This illustrates a more general principle: markets "dominate" (i.e. are eventually at least good as) all of their traders. Thus a market can be regarded as a learning mechanism.

The second work we will describe is Oesterheld's framework for "boundedly rational agents" [PM2], which, on its surface, seems to address my entire research problem: it constructs an agent that acts in a way that is "boundedly rational", specifically in the setting of a decision problem. A sketch of the construction is as given by Def 2.9.

Definition 2.9 (The problem setting)**.** A decision-problem DP is some finite set of "choices"; we deal with a decision-problem sequence $\mathsf{DP}(1), \mathsf{DP}(2), \ldots$, an agent that is a sequence of choices $x(t) \in \mathsf{DP}(t)$ and rewards $r(t) \in [0,1]$ – all of these, including the decision-problem sequence itself, can depend on one another via mutual recursion.

The specific agent type we are concerned with is an *estimating agent* comprised both of a sequence of *choices* $\mu^c(t)$ and a sequence of *reward estimates* $\mu^e(t) \in [0,1]$. In particular we have a class of estimating agents called "hypotheses" $\mathcal{A}$, which could e.g. be all polynomial-time estimating agents or all $O(g(t))$-time estimating agents for some non-decreasing $g(t)$: these will be our "traders".

Define an "allowance function" $\mu^{\mathrm{bh}} : \mathbb{N} \times \mathcal{A} \to \mathbb{Q}$ so that $\mu^{\mathrm{bh}}(t, \alpha)$ is interpreted as the budget allocated to hypothesis h at time t, satisfying the following properties: (1) for each t, only $\mu^{\mathrm{bh}}(t)$ has finite support (i.e. is non-zero for only finitely many α) (2) each hypothesis gets an infinite total allowance i.e. $\sum_t \mu^{\mathrm{bh}}(t, \alpha) = \infty$ for all α (3) the allowance distributed per round must approach zero $\sum_\alpha \mu^{\mathrm{bh}}(t, \alpha) \to 0$.

Definition 2.10 (Oesterheld agent)**.** The boundedly rational "estimating agent" can then be constructed as an auction conducted among all hypotheses, as described by the function μ that follows.

The paper then goes on to show some analogous domination results: that μ^e is not consistently an overestimate (it may still be an underestimate, because the

```
function $\mu$
    $\alpha_\$ \leftarrow 0$                                                    ▷ for all $\alpha \in \mathcal{A}$
    for $t \in \mathbb{N}$ do
        $(\alpha_*, \alpha^e_*) \leftarrow (\arg, \max)_{\alpha \in \mathcal{A}} \min(\alpha^e(t), \alpha_\$)$      ▷ pick highest bidder
        $(\mu^c(t), \mu^e(t)) \leftarrow (\alpha_*, \alpha^e_*)$                          ▷ and listen to him
        $\alpha_{*\$} \leftarrow \alpha_{*\$} + r(t) - \alpha^e_*$                        ▷ pay reward, deduct bid
        for $\alpha \in \mathcal{A}$ do
            $\alpha_\$ \leftarrow \alpha_\$ + \mu^{\mathrm{bh}}(t, \alpha)$                       ▷ give allowance
```

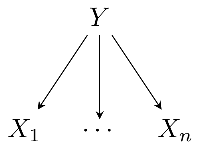

Figure 2: Causal diagram illustrating a latent variable

hypotheses are bounded by their wealth) and that every hypothesis that submits the highest estimate is tested an infinite number of times. Its key limitation, of course, is that implementing this algorithm is hopelessly intractable unless we restrict $\mathcal{A}$ to be a class of trained machine learning models (in which case it wouldn't really play the fundamental role in intelligence we would like it to).

Fundamentally similar models to [PM2] are found in [PM3–PM5]: in brief, [PM3] constructs an agent called the "Hayek machine" which specifically performs an Oesterheld-like algorithm in a "Block World" game; [PM4, PM5] implement similar algorithms in a Reinforcement Learning setting and for worlds described by partially observable Markov Decision Processes respectively. I have not read these papers in detail yet, as I only recently learned of them, but will do so subsequently.

2.3 Latent space prediction markets

Finally we discuss [L1], which raises precisely the question of latent variable prediction markets. The framework proposed is very straightforward: suppose $X_1, \ldots X_n$ are some correlated random variables (e.g. "Economic metric i will be good"). Then "we" create a latent variable Y, say with some natural language description ("The economy will be good"). Users can then submit bids either on $\mathbf{Pr}[Y]$ directly or on the distribution $\mathbf{Pr}[X_i|Y]$. Y itself will never be revealed, but the distribution $\mathbf{Pr}[X_i]$ can be inferred from these distributions, and bettors are then scored based on their implied bet on this, i.e. proportional to $\log \mathbf{Pr}[\vec{x}] - \log \mathbf{Pr}_0[\vec{x}]$.

Fig 2 illustrates the idea: postulating a latent Y constrains the space of possible joint distributions $\mathbf{Pr}[X_1, \ldots X_n]$ to only those that can be factored as $\sum_y \mathbf{Pr}[X_1|Y = y] \ldots \mathbf{Pr}[X_n|Y = y]\mathbf{Pr}[Y = y]$ for some suitable Y. This incentivizes traders to choose good latents and bet correctly on them.

However, this formalism retains several limitations: it does not specify *how* to determine which variables to specify a latent between, the specific choice of latent is maintained only by Schelling point inertia (at least in the absence of "whale" traders, which I suspect would create incentives for strategic behaviour), and it is not clear how to handle scenarios with multiple underlying latents.

3 Betting on FOL sentences

The content here is an abridged version of my paper [O1], containing only the main definitions and results. Consult the paper for a fuller discussion.

Notation. Booleans are denoted as $\{\bot, \top\}$ representing "false" and "true" respectively. Multivariable functions may be denoted as $f : S_1 \to S_2 \dots S_n \to T$ i.e. with $\to$ right-associative; we may sometimes type a function $f : S \to T$ as $f : (s : S) \mapsto T$ or even $f : s \mapsto T$ like a sort of λ-notation; wherein unless otherwise specified, time denoted by t belongs to $\mathbb{N}$. $\mathrm{pr}_{TU} : S \times T \times U \to T \times U$ denotes projection; $\operatorname{finset} S$ denotes the set of finite sets of elements in S; for a subset $T \subseteq S$, $\neg T$ denotes its complement; if a set S contains an element $\mathbf{0}$, then $f : T \to_\circ S$ denotes a function with finite support $\operatorname{supp} f := f^{-1}(\neg\{\mathbf{0}\}) := \{x \mid f(x) \neq \mathbf{0}\}$; the addition of a canonical $\mathbf{0}$ element to a set S is denoted by $\overline{S} := S \cup \{\mathbf{0}\}$. We use the notation $f : T \to_{\mathsf{c}} S$ to denote an arbitrary partial computable function, $f : T \to_{\mathsf{p}} S$ to denote a function that is polynomial-time in input t, $f : T \to_{\downarrow} S$ to denote a non-increasing function, and $f : T \to_{\_} S$ to denote a piecewise-constant function over a finite number of pieces, i.e. an S-labeled partition of T (so e.g. $f : T \to_{\_\downarrow} S$ denotes a function that is both non-increasing and piecewise-constant), and $|l|$ to denote the length of a clopen interval or finite union of clopen intervals $l \subseteq \mathbb{Q}$. Denote by string the set of finite strings with the infix $l_1 : l_2$ indicating string concatenation, Prop the set of FOL sentences in prenex normal form, $\mathrm{Prop}^+ := \mathrm{Prop} \cap \neg\Delta_0$ and $\Sigma = \bigcup \Sigma_n$, $\Pi = \bigcup \Pi_n$. For $n \in \mathbb{N}$ and $P \in \mathrm{Prop}^+$, the replacement of the leading variable in P by n is denoted $P[n]$ (i.e. if $P := \exists x, p(x)$ or $P := \forall x, p(x)$ then $P[n] := p(n)$). Let $N \in \overline{\operatorname{finset} \mathbb{N}}$: if $N = \mathbf{0}$, $P[N] := P$; else if $P \in \Sigma$, then $P[N] := \bigvee_{n \in N} P[n]$; else if $P \in \Pi$, then $P[N] := \bigwedge_{n \in N} P[n]$, all reduced to prenex normal form. We may also use this notation for a map $\breve{\alpha} : \mathrm{Prop} \to \operatorname{finset} \mathbb{N}$, i.e. denote $P[\breve{\alpha}] := \tilde{\alpha} P \breve{\alpha}(P)$, and represent successive applications as $P[M, N] := P[M][N]$, etc. likewise $P[\breve{\alpha}, \breve{\beta}] := P[\breve{\alpha}][\breve{\beta}]$ etc.

First, consider what the *ideal* solution to our problem would look like, to help pin down exactly what it is we want. The ideal "asset" for some FOL sentence P would be one that would pay off \$1 iff P were true, and \$0 otherwise (we don't need to bother with formal questions about defining a truth predicate: we simply demand that the payoff of the asset equalling \$1 is *logically equivalent* to P). But we can prove quite easily that there is no computable mechanism to define such an asset.

Lemma 3.1 (the problem is not trivial)**.** *An asset mechanism is a computable*

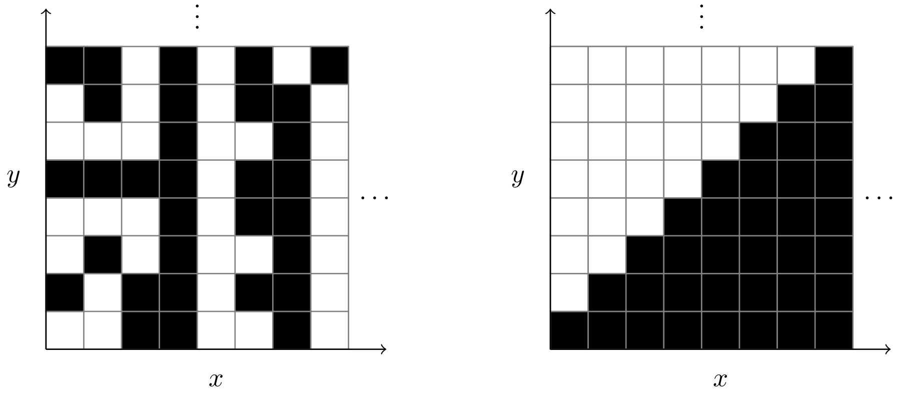

Figure 3: Shading $P(x, y)$ as a checkerboard, the Σ_2 sentence $\exists x, \forall y, P(x, y)$ can be interpreted as "there is an infinite black column". Left: an arbitrary example; Right: the example of $P(x, y) := x \geq y$, where the Σ_2 sentence is false, yet it is true for every "finite window".

real-valued function $v : \mathbb{N} \times \text{Prop} \to \mathbb{R}$ *such that* $\lim_{t\to\infty} v(t, P) = 1$ *if* P *and* 0 *if* $\neg P$. *A scoring rule mechanism is a computable real-valued function* $s : \mathbb{N} \times \text{Prop}$ *such that* $\lim_{t\to\infty} s(t, P, p) = \log(p)$ *if* P *and* $\log(1-p)$ *if* $\neg P$. *If* Prop*, the class of propositions considered, includes at least* Σ_4 *or* Π_4 *sentences, neither an asset mechanism nor a scoring rule mechanism exists.*

Proof. The statements $\lim_{t\to\infty} v(t, P) = 1$ and $\lim_{t\to\infty} s(t, P, p) = \log(p)$ are both Π_3 for any P, p; thus for them to be equivalent to P for all P would violate Tarski's theorem. □

Fig 3 illustrates an example of a "sophomore's dream" solution to our problem, which fails.

Instead in this paper, we propose a game-theoretic solution relying on a slightly modified version of Hintikka's Verification-Falsification (VF) game [0], the crucial difference being that players' moves are in $\text{finset}\,\mathbb{N}$ rather than $\mathbb{N}$ itself.

Definition 3.1 (Verification-Falsification game)**.** To each FOL sentence P we associate a game between two players, the "Verifier" and the "Falsifier". If P is Δ_0, then the Verifier or Falsifier win \$1 if P is true or false respectively. Else if P is Σ, the Verifier goes first and if P is Π, the Falsifier goes first. The first player chooses $S \in \text{finset}\,\mathbb{N}$ as their first move, and the game proceeds as the VF game for $P[S]$.

The idea is then as follows: *we measure how much traders are willing to pay to play the Verification-Falsification game for some sentence* P. Another way of expressing this is: the asset for some sentence $\exists x, P(x)$ is the *option* to exchange it for any $\bigvee_{x \in S} P(x)$ for S finite; while $\forall x, P(x)$ is the *obligation* to exchange it for any $\bigwedge_{x \in S} P(x)$ chosen by the opponent.

As we know from 3.1, this framework does not have the standard property that agents are incentivized to bet sentences up to their subjective probability. In particular, we see that an agent's fair price in this game not only depends on its belief in the statement, but its ability to construct specific values for the bound variables to win the game. Still, we may construct a theoretical program market for our framework that has at least some closely related desirable properties.

Definition 3.2 (Prediction Market for VF games)**.** Define the following:

- A *price setter* π is composed of:
  - a *price sequence* $\pi^* : t \mapsto_{\mathsf{p}} \text{Prop} \to_{\circ} [0,1]$
  - a *player* $\tilde{\pi} : t \mapsto_{\mathsf{p}} \text{Prop} \to_{\circ} \text{string} \to_{\circ} \overline{\text{finset}\, \mathbb{N}}$
  - a *labeler* $\pi^{\#} : t \mapsto_{\mathsf{p}} \text{Prop} \to_{\circ} \mathbb{Q} \to_{-} \overline{\text{string}}$ such that $\left|\text{supp}\, \pi^{\#}(t,P)\right| = \mu_{\$}(t,P)$ (defined by mutual recursion below)
- An *agent* α is composed of:
  - an *endowment* $\alpha^{\text{b}} \in \mathbb{Q}$ and a *birthday* $\alpha^{\text{t}} \in \mathbb{N}$
  - a *trader* $\hat{\alpha} : t \mapsto_{\mathsf{p}} \text{Prop} \to_{\circ} [0,1] \to_{-\downarrow} \mathbb{Q}$
  - a *player* $\tilde{\alpha} : t \mapsto_{\mathsf{p}} \text{Prop} \to_{\circ} \text{string} \to_{\circ} \overline{\text{finset}\, \mathbb{N}}$
  - a *labeler* $\alpha^{\#} : t \mapsto_{\mathsf{p}} \text{Prop} \to_{\circ} \mathbb{Q} \to_{-} \overline{\text{string}}$ such that $\left|\text{supp}\, \alpha^{\#}(t,P)\right| = \alpha_{\$}(t,P)$ (defined by mutual recursion below)
  - an *inventory* $\alpha_{\$} : t \mapsto_{\mathsf{p}} \text{Prop} \to_{\circ} \mathbb{Q}$ defined by (initial conditions) $\alpha_{\$}(s,\top) = 0$ for $s < \alpha^{\text{t}}$, $\alpha_{\$}(\alpha^{\text{t}},\top) = \alpha^{\text{b}}$ and for all other propositions $\alpha_{\$}(0,P) = 0$ and (recursion rule) $\alpha_{\$}(t+1) = \alpha_{\$}(t) + \sum_{P,i} \delta\alpha_{\$}^{P,i}(t)$ where:
    - orders placed: $\delta\alpha_{\$}^{P,1}(t,P) = \lambda\hat{\alpha}(t,P,\pi^*(t,P))$ where λ is 1 if the following conditions are met (else 0): for all P, $\max_p -\hat{\alpha}(t,P,p) \leq \alpha_{\$}(t,P)$ (you're not selling what you don't have) and $\sum_P \max_p p\hat{\alpha}(t,P,p) \leq \alpha_{\$}(t,\top)$ (you can afford all your purchases)
    - cost of orders placed: $\delta\alpha_{\$}^{P,2}(t,\top) = -\pi^*(t,P)\delta\alpha_{\$}^{P,1}(t,P)$
    - the moves played (if $P \in \Sigma$): $\delta\alpha_{\$}^{P,3}(t,P) = -\sum_{l \in \text{supp}\, \tilde{\alpha}(t,P)} \left|\alpha^{\#}(t,P)^{-1}(l)\right|$ and $\delta\alpha_{\$}^{P,4}(t,P[\tilde{\alpha}(t,P,l)]) = \left|\alpha^{\#}(t,P)^{-1}(l)\right|$
    - the opponent's moves (if $P \in \Pi$): $\delta\alpha_{\$}^{P,5}(t,P) = -\sum_{l \in \text{supp}\, \tilde{\alpha}(t,P)} \left|\pi^{\#}(t,P)^{-1}(l)\right|$ and $\delta\alpha_{\$}^{P,6}(t,P[\tilde{\pi}(t,P,l)]) = \left|\pi^{\#}(t,P)^{-1}(l)\right|$
    - the payout from empirical truth (if $P \in \Delta_0$): if $P \in \text{supp}\, \xi(t)$, then $\delta\alpha_{\$}^{P,7}(t,P) = -\alpha_{\$}(t,P)$ and $\delta\alpha_{\$}^{P,8}(t,\xi(t,P)) = \alpha_{\$}(t,P)$

- the *empirical reality* is a process $\xi : t \mapsto_{\mathsf{p}} \Delta_0 \to_\circ \overline{\{\bot, \top\}}$ such that $s \leq t, P \in \operatorname{supp} \xi(s) \implies \xi(t, P) = \xi(s, P)$ and $\bigcup_t \operatorname{supp} \xi(t) = \Delta_0$ and $\xi(t, \neg P) = \neg \xi(t, P)$

- The type of agents is denoted as $\mathcal{A} := \mathcal{A}^{\mathrm{b}} \times \mathcal{A}^{\mathrm{t}} \times \hat{\mathcal{A}} \times \tilde{\mathcal{A}} \times \mathcal{A}^{\#}$ where the respective types are already as specified (including sub-typing by the requisite condition); define specifically $\mathcal{A}^{\circ} := \hat{\mathcal{A}} \times \tilde{\mathcal{A}} \times \mathcal{A}^{\#}$.

- The *father of agents* is an enumerator and allocator for agents, i.e. a map $\mu^{\mathrm{bh}} : t \mapsto \mathcal{A}$ such that $\mathrm{pr}_{\mathcal{A}^{\circ}} \circ \mu^{\mathrm{bh}}$ is bijective; the total endowment is finite $\sum_t \mu^{\mathrm{bh}}(t)^{\mathrm{b}} < \infty$; and the birthdays are correct $\mu^{\mathrm{bh}}(t)^{\mathrm{t}} = t$. It is associated with an *aggregate agent* as follows:

  - $\mu^{\mathrm{b}} = \sum_t \mu^{\mathrm{bh}}(t)^{\mathrm{b}}$ and $\mu^{\mathrm{t}} = 0$
  - $\hat{\mu}(t) = \sum_{\alpha = \mu^{\mathrm{bh}}(1), \dots \mu^{\mathrm{bh}}(t)} \lambda_\alpha \hat{\alpha}(t)$ where λ_α is 1 if the following conditions are met (else 0): for all P, $\max_p -\hat{\alpha}(t, P, p) \leq \alpha_\$(t, P)$ and $\sum_P \max_p p\hat{\alpha}(t, P, p) \leq \alpha_\$(t, \top)$ (this sum is only a finite sum, because only finitely many agents have any cash at t)
  - $\tilde{\mu}(t, P, \langle\alpha\rangle : l) = \tilde{\alpha}(t, P, l)$ where $\langle\alpha\rangle$ indicates some encoding for agents
  - $\mu^{\#}(t, P, q) = \begin{cases} \mu^{\mathrm{bh}}(t_q) : \mu^{\mathrm{bh}}(t_q)^{\#}(t, P) & \text{if } t_q \leq t \\ \mathbf{0} & \text{else} \end{cases}$

    where t_q is the smallest value such that $\sum_{s \leq t_q} \mu^{\mathrm{bh}}(s)_\$(t, P) \geq q$, if it exists, else ∞.

- A special price setter ϖ, called *equilibrium*, is defined as follows – for each t and P:

  - $\varpi^*(t, P)$ is any solution p to $\hat{\mu}(t, P, p) - \hat{\mu}(t, \neg P, 1 - p) = 0$.
  - $\tilde{\varpi}(t, P) = \tilde{\mu}(t, \neg P)$
  - $\varpi^{\#}(t, P) = \mu^{\#}(t, \neg P)$

A more readable description: each agent manages its *inventory* $\alpha_\$(t) : \mathrm{Prop} \to_\circ \mathbb{Q}$; in particular $\alpha_\$(t, \top)$ denotes an agent's cash reserves. At each point in time, $\hat{\alpha}(t, P)$ denotes its demand (or supply, if negative) schedule for P, which is a sum of limit orders. The description of the player has some subtle considerations: (1) the moves are in $\overline{\operatorname{finset} \mathbb{N}}$ rather than $\operatorname{finset} \mathbb{N}$, because the move $\mathbf{0}$ is interpreted as "pass", i.e. should the agent wishes to not instantly play his move when he acquires P but compute his move over several time-steps (2) an agent that holds multiple stocks of some sentence might not want to play the same move on all of them.

We capture this behaviour by having the agent *labelling* different portions of his inventory of stock P with different labels, i.e. a labelled partition $\mathbb{Q} \to_{-} \overline{\text{string}}$ of $\mathbb{Q}$ so that only a fragment of $\mathbb{Q}$ whose length equals the agent's stock in P is mapped to a label, everything else is mapped to $\mathbf{0}$. When $\tilde{\alpha}$ then plays

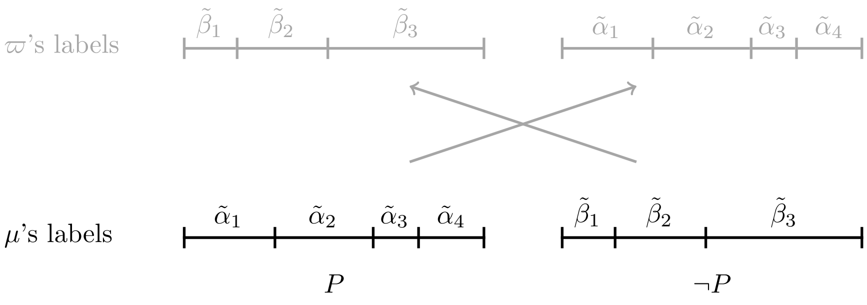

Figure 4: Illustration of ϖ's player countering μ

a move for sentence P and label l, the amount of P stocks labelled with l will be removed from the inventory and replaced with $P[\tilde{\alpha}(t, P, l)]$.

In particular, the aggregate agent μ uses this system to keep the inventories for different agents separate by prefixing their labels with some code for each agent as $\langle\alpha\rangle : l$ – this is necessary for the crux of the program market concept, which is that each agent can only trade with its own money, so more successful agents gain influence (in the form of wealth) while those that go bankrupt are effectively removed from the market. This is done through the indicator denoted by λ_α, which checks if α can afford the trades it makes.

Finally, some notes regarding equilibrium calculation: in our framework, agents provide independent demand schedules for each sentence (i.e. a map $\text{Prop} \to_\circ ([0,1] \to \mathbb{Q})$, rather than a map $(\text{Prop} \to_\circ [0,1]) \to \mathbb{Q}$): in other words, should the agents wish for cross-elasticity in their demand-schedules, the onus is on them to estimate the prices of other sentences. This reduces equilibrium calculation to finding the zero of a non-increasing piecewise-constant function, which is elementary. The equilibrium price setter's player is more subtle: at equilibrium price, μ buys equal amounts of P and $\neg P$, however it may use different players for different portions of each. So we have $\tilde{\varpi}$ use μ's players for $\neg P$ against μ's players for P and vice versa so it wins exactly as many games as it loses, as illustrated in Fig 4.

Definition 3.3 (Exploitation)**.** An agent α is said to *exploit* a price-setter π if $\{\alpha_\$(t, \top) : t \in \mathbb{N}\}$ is bounded from below but not bounded from above.

Lemma 3.2 (Inexploitabity)**.** *There is no agent $\alpha \in \mathcal{A}$ that exploits ϖ.*

Proof. If any $\alpha \in \mathcal{A}$ exploits ϖ, then so does μ (because the inventory allocated to α will increase without bound, while all other agents' inventories are bounded below by 0 since they can only spend from their alloted inventory). However, by construction, μ does not exploit ϖ (as it always holds an equal number of P and $\neg P$ stocks, and wins exactly as many games as it loses). Thus, no $\alpha \in \mathcal{A}$ exploits ϖ. □

Theorem 3.3 (Convergence)**.** $\lim_{t\to\infty} \varpi^*(t, P)$ *exists for all P; denote this as* $\varpi^\infty(P)$*.*

Proof. Suppose it didn't; then there exists $x \in (0,1)$ and $\varepsilon > 0$ such that $\varpi^*(t,P) > x + \varepsilon$ infinitely often and $\varpi^*(t,P) < x - \varepsilon$ infinitely often. Then consider an agent given by a trader that sells when $\varpi^*(t,P) > x+\varepsilon$ and buys when $\varpi^*(t,P) < x - \varepsilon$, and a trivial player (doesn't play at all; returns **0** each time). This agent exploits the market. □

Ideally, we would like to say that our market learns to correctly price very FOL sentence: that P is equivalent to $\varpi^\infty(t,P) = 1$, and $\neg P$ is equivalent to $\varpi^\infty(t,P) = 0$: because if a true sentence did not approach \$1, an agent could buy arbitrarily large quantities of it and win the VF game on them. However, we know from Lemma 3.1 that this is impossible: indeed, the flaw in our intuition is that the mere existence of a winning strategy (i.e. the "truth" of P) does not imply it is actually computable. Instead, we adopt the following "constructivist" notion of truth, closely related to the notion of "CGTS-truth" ("Computational Game Theoretic Semantics Truth") proposed in [M1].

Definition 3.4 (Constructive truth)**.** Denote $\breve{\mathcal{A}} := \mathrm{Prop} \to_{\mathsf{c}} \mathrm{finset}\,\mathbb{N}$. For any $P \in \mathrm{Prop}$ and $\breve{\alpha}, \breve{\beta} \in \breve{\mathcal{A}}$, observe that the sequence $P[\breve{\alpha}, \breve{\beta}, \breve{\alpha}, \breve{\beta}, \dots]$ eventually converges to a Δ_0 sentence: denote this sentence by $\mathrm{Game}(P, \breve{\alpha}, \breve{\beta})$. Then an FOL-sentence P is said to be $\breve{\mathcal{A}}$-true if $\exists \breve{\alpha} \in \breve{\mathcal{A}}, \forall \breve{\beta} \in \breve{\mathcal{A}}, \mathrm{Game}(P, \breve{\alpha}, \breve{\beta})$, and $\breve{\mathcal{A}}$-false if $\exists \breve{\beta} \in \breve{\mathcal{A}}, \forall \breve{\alpha} \in \breve{\mathcal{A}}, \neg\,\mathrm{Game}(P, \breve{\alpha}, \breve{\beta})$.

Lemma 3.4 (Correspondence between $\breve{\mathcal{A}}$ and $\tilde{\mathcal{A}}$)**.** *For any* $\breve{\alpha} : \mathrm{Prop} \to_{\mathsf{c}} \mathrm{finset}\,\mathbb{N}$ *there is a corresponding* $\tilde{\alpha} : t \mapsto_{\mathsf{p}} \mathrm{Prop} \to_{\circ} \mathrm{string} \to_{\circ} \overline{\mathrm{finset}\,\mathbb{N}}$ *that executes it,, such that the following conditions are met: (1)* $\forall s \neq t, \mathrm{supp}\,\breve{\alpha}(t) \cap \mathrm{supp}\,\breve{\alpha}(s) = \varnothing$ *and (2) if* $P \in \mathrm{supp}\,\breve{\alpha}$ *(*$\breve{\alpha}$ *halts on input* P*) then* $\exists t, \tilde{\alpha}(t, P, \texttt{``foo''}) = \breve{\alpha}(P)$ *else* $\forall t, \tilde{\alpha}(t, P, \texttt{``foo''}) = \mathbf{0}$.

Proof. Fix an enumeration of Prop i.e. a bijection $\zeta : \mathbb{N} \to \mathrm{Prop}$, and fix a model of computation for $\breve{\alpha}$, e.g. a Turing Machine. Then define $\tilde{\alpha}$ as follows:

$$\tilde{\alpha}(t, P, l) = \begin{cases} \breve{\alpha}(P) & \text{if } P \in \{\zeta(0), \dots, \zeta(t)\}, \\ & \quad \mathrm{Halts}(\breve{\alpha}, P, t), \\ & \quad l = \texttt{``foo''} \\ \mathbf{0} & \text{else} \end{cases}$$

□

Theorem 3.5 (Learning constructive truth)**.** *If* P *is an* $\breve{\mathcal{A}}$*-true sentence, then* $\varpi^\infty(P) = 1$.

Proof. Suppose it wasn't; then there is some ε such that $\varpi^*(t,P)$ is below $1 - \varepsilon$ infinitely many times. Then consider the agent given by (1) the trader $\hat{\alpha}$ that buys whenever $\varpi^*(t,P) < 1 - \varepsilon$ (2) the player $\tilde{\alpha}$ that is the Lemma 3.4-correspondent of the $\breve{\alpha}$ affirmed in the hypothesis $\exists \breve{\alpha} \in \breve{\mathcal{A}}, \forall \breve{\beta} \in \breve{\mathcal{A}}, \mathrm{Game}(P, \breve{\alpha}, \breve{\beta})$ (3) the labeler $\alpha^\#$ that returns `''foo''` on all outputs. This agent exploits the market. □

Corollary 3.6 (Learning constructive falsehood)**.** *If P is an $\breve{\mathcal{A}}$-false sentence, then $\varpi^\infty(P) = 0$.*

Proof. Apply Thm 3.5 to $\neg P$. □

4 Concluding remarks

4.1 Future work

There are several research directions available to me based on the work I've done so far, broadly categorized as follows:

1. **The main project** – i.e. latent space markets, as described as RQ2 in the introduction.

2. **Interpretability** – i.e. mechanisms to incentivize agents to make bets on each others' latent spaces, as described as RQ3 in the introduction and as motivated by the discussion on [L2].

3. **Coherent extrapolated volition (CEV)** – One crucial observation underlying my project is that program markets naturally capture the notion of *rationality conditioned on algorithmic information.* [BR1] asks what an agent would look like with "unlimited algorithmic information"; program markets may be the right framework to model this problem.

4. **Practical markets for FOL sentences** – Building real-world prediction markets for FOL sentences (e.g. prediction markets for various practically relevant statistical questions may be expressed in terms of limiting distributions of experimental results, thus as Π_3 sentences). With practical markets, however, special attention must be taken to take into account asymmetries in the computational costs of players for opposing sides: in our theoretical framework, every possible player was enumerated in the market, however practically some players may be "more expensive" than others, and this may influence how much traders are willing to pay for a side. For example, this may be addressed by "separating out" the market for players from the prediction market, so that traders are explicitly paying a price for purchasing players and this price can be added to the probability estimate for a sentence. In particular, this might also make it feasible to implement such systems with an automated market maker (e.g. logarithmic market scoring [MI2, MI3]), as the obstacle to doing this immediately with our framework is that it's not obvious what player such a market-maker would use.

5. **Implications for mathematical logic** – One potential implication of my work in mathematical logic: it provides a new approach to measuring the strength of a mathematical theory: in a slightly modified version of game semantics that allows one player to "go back and change its moves"

[M2], a formal proof from a mathematical theory may be understood as a computable strategy to this modified verification-falsification game [M1]; we my then consider the maximum wealth that can be acquired by an agent that only trades and plays in accordance with a theory, and regard this as a measure of the strength of the theory.

6. **Game-theoretic probability** – Our result (Thm 3.5) is still somewhat weak, in that it says nothing about sentences that are neither $\breve{\mathcal{A}}$-true nor $\breve{\mathcal{A}}$-false – from Thm 3.3, we know that the market price of every sentence converges to *something*, and it would be interesting to study the nature of this limiting distribution. It would not be a probability distribution in the traditional sense – i.e. you would not have $\mathbf{Pr}[\exists x, P(x)] = \sup_x \mathbf{Pr}[\exists i \leq x, P(i)]$ and $\mathbf{Pr}[\forall x, P(x)] = \inf_x \mathbf{Pr}[\forall i \leq x, P(i)]$, as this would contradict Lemma 3.1. Instead we might consider an alternate definition of a σ-algebra which is required to be closed only under unions and intersections of *computable sequences*, i.e. replacing the countable union axiom with "$\psi : \mathbb{N} \to_{\mathsf{c}} \mathcal{F} \implies \bigcup_i \psi(i) \in \mathcal{F}$" and adopting a suitable generalized notion of probability measure on this algebra. Possibly relevant work to this end includes: quantifier algebras [0], Shafer & Vovk's game-theoretic formulation of probability theory [M3, M4] and Japaridze's "Computability Logic" [M5, M6].
7. **Bridges to neural networks** – as it stands, program markets remain quite impractical; while I believe latent space markets will address this in part, it may also be worth investigating if any of our results could translate to existing AI architectures; [PM6], which draws analogies between markets and neural networks, might be worth studying to this end.

Of these, I should have the first two fully fleshed-out in a paper by the next annual review; the remaining are side-quests which I can initiate some work in, perhaps with some collaborators.

4.2 Reflections

Related events. I attended the Co-operative AI Foundation's summer workshop in July 2023, where I presented a brief poster on my ideas and discussed them with some people working on related fields.

I'll probably submit my first paper to LOFT 2024 (Jul) or WINE 2024 (Dec) after a finishing touch. I'm also considering attending workshops like SERI MATS and UC Berkeley SPAR to find potential collaborators, as my research is starting to open up several different opportunities that I cannot simultaneously focus on all at once.

Professional development and transferable skills. Generally speaking it has been quite exciting to see how closely my specific topic is related to so many other areas of interest to me, and my work has helped me gain a much more intimate familiarity with the literature in these areas. Two particular "soft skills" I've gotten better at are literature review and academic networking (the latter over the course of the CAIF workshop), two areas I've always been kinda sucky at.

Collaboration with supervisor and others. I generally have weekly meetings with my supervisor, and resort to emails when that is not possible. I do not have other collaborators as of yet, though I am seeking to expand on that.

Bibiliography

Papers (drafts and publications)

My framework for betting on FOL sentences [O1].

[O1] Abhimanyu Pallavi Sudhir. *Betting on What Is Neither Verifiable nor Falsifiable.* Dec. 1, 2023. preprint.

Markets and information

General introductions to prediction markets [MI1–MI3].

[MI1] Vincent Conitzer. *Prediction Markets, Mechanism Design, and Cooperative Game Theory.* May 9, 2012. DOI: `10.48550/arXiv.1205.2654`. arXiv: `1205.2654 [cs]`. URL: `http://arxiv.org/abs/1205.2654` (visited on 06/22/2023). preprint.

[MI2] Robin Hanson. "Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation". In: *The Journal of Prediction Markets* 1.1 (Jan. 2002), pp. 3–15. DOI: `10.5750/jpm.v1i1.417`.

[MI3] Robin Hanson. "Combinatorial Information Market Design". In: *Information Systems Frontiers* 5.1 (Jan. 1, 2003), pp. 107–119. ISSN: 1572-9419. DOI: `10.1023/A:1022058209073`. URL: `https://doi.org/10.1023/A:1022058209073` (visited on 05/05/2023).

Program markets

[PM1, PM2] are the classical works on this; [PM6] describes a relationship between markets and neural networks; [PM3–PM5] are similar works to [PM2].

[PM1] Scott Garrabrant et al. *Logical Induction.* Sept. 12, 2016. arXiv: `1609.03543`. URL: `http://arxiv.org/abs/1609.03543`. preprint.

[PM2] Caspar Oesterheld, Abram Demski, and Vincent Conitzer. "A Theory of Bounded Inductive Rationality". In: *Electronic Proceedings in Theoretical Computer Science* 379 (July 11, 2023), pp. 421–440. ISSN: 2075-2180. DOI: `10.4204/EPTCS.379.33`. arXiv: `2307.05068 [cs]`. URL: `http://arxiv.org/abs/2307.05068` (visited on 12/01/2023).

[PM3] Eric B. Baum. "Toward a Model of Intelligence as an Economy of Agents". In: *Machine Learning* 35.2 (May 1, 1999), pp. 155–185. ISSN: 1573-0565. DOI: 10.1023/A:1007593124513. URL: https://doi.org/10.1023/A:1007593124513 (visited on 08/26/2023).

[PM4] Michael Chang et al. *Decentralized Reinforcement Learning: Global Decision-Making via Local Economic Transactions.* Aug. 14, 2020. DOI: 10.48550/arXiv.2007.02382. arXiv: 2007.02382 [cs, stat]. URL: http://arxiv.org/abs/2007.02382 (visited on 07/19/2023). preprint.

[PM5] Ivo Kwee, Marcus Hutter, and Juergen Schmidhuber. *Market-Based Reinforcement Learning in Partially Observable Worlds.* Version 1. May 15, 2001. DOI: 10.48550/arXiv.cs/0105025. arXiv: cs/0105025. URL: http://arxiv.org/abs/cs/0105025 (visited on 12/01/2023). preprint.

[PM6] John Wentworth. *Competitive Markets as Distributed Backprop.* Nov. 10, 2018. URL: https://www.lesswrong.com/posts/brhWPoNsBN7za3xjs/competitive-markets-as-distributed-backprop (visited on 04/23/2023). preprint.

Latent space

[L2] lays out the "Eliciting Latent Knowledge" program for interpetability; the topic of prediction markets for latent space knowledge has been partially addressed in [L1, L3, L4]; Wentworth's work on natural abstractions [L5, L6] can be understood as introducing a logic or algebra for the latent space.

[L1] tailcalled. *Latent Variables for Prediction Markets: Motivation, Technical Guide, and Design Considerations.* LessWrong. Feb. 12, 2023. URL: https://www.lesswrong.com/posts/ufW5LvcwDuL6qjdBT/latent-variables-for-prediction-markets-motivation-technical (visited on 06/26/2023).

[L2] Paul Christiano, Mark Xu, and Ajeya Cotra. *Eliciting Latent Knowledge.* Dec. 2021. URL: https://docs.google.com/document/d/1WwsnJQstPq91_Yh-Ch2XRL8H_EpsnjrC1dwZXR37PC8/edit?usp=embed_facebook (visited on 12/02/2023). preprint.

[L3] Aurélien Baillon. "Bayesian Markets to Elicit Private Information". In: *Proceedings of the National Academy of Sciences* 114.30 (July 25, 2017), pp. 7958–7962. DOI: 10.1073/pnas.1703486114. URL: https://www.pnas.org/doi/full/10.1073/pnas.1703486114 (visited on 06/11/2023).

[L4] Aurélien Baillon and Yan Xu. "Simple Bets to Elicit Private Signals". In: *Theoretical Economics* 16.3 (2021), pp. 777–797. ISSN: 1555-7561. DOI: 10.3982/TE4343. URL: https://onlinelibrary.wiley.com/doi/abs/10.3982/TE4343 (visited on 05/27/2023).

[L5] John Wentworth and David Lorell. *Some Rules for an Algebra of Bayes Nets*. LessWrong. Nov. 16, 2023. URL: https://www.lesswrong.com/posts/XHtygebvHoJSSeNPP/some-rules-for-an-algebra-of-bayes-nets (visited on 01/15/2024).

[L6] johnswentworth and David Lorell. *Natural Latents: The Math*. LessWrong. Dec. 27, 2023. URL: https://www.lesswrong.com/posts/dWQWzGCSFj6GTZHz7/natural-latents-the-math (visited on 01/15/2024).

Math

Related math and formal logic I've consulted over the course of my work on FOL markets: [M7] on algorithmic complexity; [M3, M4] discuss an alternate "game-theoretic" formulation of probability theory, something of that sort might be necessary to capture long-run prices of FOL sentences in my framework; [0, M1, M2, M5, M6] address game semantics and computability logic.

[M1] Julien Boyer and Gabriel Sandu. "Between Proof and Truth". In: *Synthese* 187.3 (Aug. 1, 2012), pp. 821–832. ISSN: 1573-0964. DOI: 10.1007/s11229-011-9903-y. URL: https://doi.org/10.1007/s11229-011-9903-y (visited on 11/28/2023).

[M2] Denis Bonnay. "Preuves et Jeux Sémantiques". In: *Philosophia Scientiae* 8.2 (2004), pp. 105–123. ISSN: 1775-4283. URL: http://www.numdam.org/item/PHSC_2004__8_2_105_0/ (visited on 11/28/2023).

[M3] Glenn Shafer and Vladimir Vovk. *Probability and Finance: It's Only a Game!* John Wiley & Sons, Feb. 25, 2005. 438 pp. ISBN: 978-0-471-46171-5.

[M4] Glenn Shafer and Vladimir Vovk. *Game-Theoretic Foundations for Probability and Finance*. John Wiley & Sons, May 29, 2019. 480 pp. ISBN: 978-0-470-90305-6.

[M5] Giorgi Japaridze. "In the Beginning Was Game Semantics". In: *Games: Unifying Logic, Language and Philosophy*. Berlin: Springer Verlag, 2009, pp. 249–350. ISBN: 978-1-4020-9373-9. DOI: 10.1007/978-1-4020-9374-6_11. arXiv: cs/0507045. URL: http://arxiv.org/abs/cs/0507045 (visited on 09/16/2023).

[M6] Giorgi Japaridze. *Survey of Computability Logic*. 2015. URL: http://www.csc.villanova.edu/~japaridz/CL/ (visited on 09/16/2023).

[M7] Shyam Wuppuluri and Francisco Antônio Doria. *Unravelling Complexity: The Life and Work of Gregory Chaitin*. World Scientific Publishing Company Pte Limited, Dec. 14, 2019. 444 pp. ISBN: 9789811200069. Google Books: C1_YwAEACAAJ.

Economics

General discussion of computational costs in economics [E1–E3]; on transaction costs [E4].

[E1] Alfred Lorn Norman. "Computability, Complexity and Economics". In: *Comput. Econ.* 7.1 (Feb. 1994), pp. 1–21.

[E2] John P Rust. "Dealing with the Complexity of Economic Calculations". In: *SSRN Electron. J.* (1997).

[E3] P. Chen. "On the Efficiency and Complexity of Computational and Economic Processes". PhD thesis. USA: Northwestern University, 1990.

[E4] Yoram Barzel. "Transaction Costs: Are They Just Costs?" In: *Zeitschrift für die gesamte Staatswissenschaft / Journal of Institutional and Theoretical Economics* 141.1 (1985), pp. 4–16. ISSN: 00442550. JSTOR: 40750776. URL: http://www.jstor.org/stable/40750776 (visited on 01/25/2023).

Bounded rationality

General overviews and discussion of bounded rationality [BR2–BR5]; Russell & Subramanian's "bounded optimality" [BR6, BR7]; Coherent Extrapolated Volition [BR1]; equilibriums of games between machines/programs [BR8–BR12]; thermodynamic rationality [BR13–BR16].

[BR1] Eliezer Yudkowsky. *Coherent Extrapolated Volition.* 2004. URL: https://intelligence.org/files/CEV.pdf. preprint.

[BR2] Herbert Simon. "Models of Man: Social and Rational". In: New York: John Wiley and Sons, 1957.

[BR3] Gerd Gigerenzer and Reinhard Selten, eds. *Bounded Rationality: An Adaptive Toolbox.* Dahlem Workshop Reports. London, England: MIT Press, July 2002.

[BR4] Gerd Gigerenzer. "Towards a Rational Theory of Heuristics". In: *Minds, Models and Milieux.* Palgrave Macmillan UK, 2016, pp. 34–59. DOI: 10.1057/9781137442505_3. URL: https://doi.org/10.1057/9781137442505_3.

[BR5] Kenneth J. Arrow. "Is Bounded Rationality Unboundedly Rational? Some Ruminations." In: Models of a Man: Essays in Memory of Herbert A. Simon. Cambridge, MA, US: MIT Press, 2004, pp. 47–55. ISBN: 0-262-01208-1.

[BR6] Stuart J. Russell and Devika Subramanian. "Provably Bounded-Optimal Agents". In: *CoRR* cs.AI/9505103 (1995). URL: https://arxiv.org/abs/cs/9505103.

[BR7] Stuart Russell. "Rationality and Intelligence: A Brief Update". In: *Fundamental issues of artificial intelligence* (2016), pp. 7–28.

[BR8] Joseph Y. Halpern and Rafael Pass. "Algorithmic Rationality: Game Theory with Costly Computation". In: *CoRR* abs/1412.2993 (2014). arXiv: 1412.2993. URL: http://arxiv.org/abs/1412.2993.

[BR9] Joseph Y. Halpern and Rafael Pass. "I Don't Want to Think About It Now:Decision Theory With Costly Computation". In: *CoRR* abs/1106.2657 (2011). arXiv: `1106.2657`. url: `http://arxiv.org/abs/1106.2657`.

[BR10] Moshe Tennenholtz. "Program Equilibrium". In: *Games and Economic Behavior* 49.2 (Nov. 2004), pp. 363–373. doi: `10.1016/j.geb.2004.02.002`. url: `https://doi.org/10.1016/j.geb.2004.02.002`.

[BR11] Richard L. Lewis, Andrew Howes, and Satinder Singh. "Computational Rationality: Linking Mechanism and Behavior Through Bounded Utility Maximization". In: *Topics in Cognitive Science* 6.2 (2014), pp. 279–311. doi: `10.1111/tops.12086`.

[BR12] Shlomo Zilberstein. "Metareasoning and Bounded Rationality". In: *Metareasoning*. The MIT Press, Mar. 2011, pp. 27–40. doi: `10.7551/mitpress/9780262014809.003.0003`. url: `https://doi.org/10.7551/mitpress/9780262014809.003.0003`.

[BR13] Pedro A. Ortega et al. *Information-Theoretic Bounded Rationality*. arXiv, 2015. doi: `10.48550/ARXIV.1512.06789`. url: `https://arxiv.org/abs/1512.06789`.

[BR14] Pedro A. Ortega and Daniel A. Braun. "Information, Utility & Bounded Rationality". In: *CoRR* abs/1107.5766 (2011). arXiv: `1107.5766`. url: `http://arxiv.org/abs/1107.5766`.

[BR15] Pedro A Ortega and Daniel A Braun. "Thermodynamics as a Theory of Decision-Making with Information-Processing Costs". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 469.2153 (2013), p. 20120683. url: `https://arxiv.org/abs/1204.6481`.

[BR16] Sebastian Gottwald and Daniel Braun. "Bounded Rational Decision-Making from Elementary Computations That Reduce Uncertainty". In: *Entropy* 21.4 (Apr. 2019), p. 375. doi: `10.3390/e21040375`. url: `https://doi.org/10.3390/e21040375`.

Lampposts for support

Literature on relatively efficient outcomes arising from "zero-intelligence" traders, hinting in favour of my intuition that a market composed of many simple traders can lead to emergent intelligence [X1–X5].

[X1] Dhananjay K. Gode and Shyam Sunder. "Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality". In: *Journal of Political Economy* 101.1 (Feb. 1993), pp. 119–137. doi: `10.1086/261868`. url: `https://doi.org/10.1086/261868`.

[X2] D. K. Gode and S. Sunder. “What Makes Markets Allocationally Efficient?” In: *The Quarterly Journal of Economics* 112.2 (May 1997), pp. 603–630. DOI: 10.1162/003355397555307. URL: https://doi.org/10.1162/003355397555307.

[X3] Karim Jamal, Michael S. Maier, and Shyam Sunder. “Simple Agents, Intelligent Markets”. In: *SSRN Electronic Journal* (2015). DOI: 10.2139/ssrn.2478665. URL: https://doi.org/10.2139/ssrn.2478665.

[X4] David I. Laibson and Leeat Yariv. “Safety in Markets: An Impossibility Theorem for Dutch Books”. In: *Levine’s Bibliography* (2007).

[X5] Alan Schwartz. “How Much Irrationality Does the Market Permit?” In: *The Journal of Legal Studies* 37.1 (Jan. 2008), pp. 131–159. DOI: 10.1086/519963. URL: https://doi.org/10.1086/519963.