Expressible inspections

Tai Wei Hu
Kellogg School of Management, Northwestern University

Eran Shmaya
Kellogg School of Management, Northwestern University

A decision maker needs predictions about the realization of a repeated experiment in each period. An expert provides a theory that, conditional on each finite history of outcomes, supplies a probabilistic prediction about the next outcome. However, there may be false experts who have no knowledge of the data-generating process and who deliver theories strategically. Hence, empirical tests for predictions are necessary. A test is manipulable if a false expert can pass the test with a high probability. Like contracts, tests have to be computable to be implemented. Considering only computable tests, we show that there is a test that passes true experts with a high probability yet is not manipulable by any computable strategy. In particular, the constructed test is both prequential and future-independent. Alternatively, any computable test is manipulable by a strategy that is computable relative to the halting problem. Our conclusion overturns earlier results that prequential or future-independent tests are manipulable, and shows that computability considerations have significant effects in these problems.

Keywords. Computability, expert testing, calibration tests, zero-sum games.

JEL classification. C44, D81, D83.

1. Introduction

Forecasting is crucial for economic planning. In many cases, such as weather or macroeconomic variables, the underlying data-generating process is stochastic instead of deterministic. Indeed, probabilistic predictions have been widely adopted in forecasting precipitation and economic variables. Although it is easy to test deterministic forecasts empirically, it is less obvious whether a stochastic forecast is useful or even meaningful. There is a literature that frames this problem in the context of testing experts: does there exist a test, which specifies situations where the expert is rejected, such that a true expert who knows the underlying process is not rejected by the test while a false expert who has no such knowledge but can make predictions strategically is rejected?

The answer to this question depends on the nature of the test. In the most general setup, the expert is required, before any outcome is realized, to provide predictions for

Tai Wei Hu: t-hu@kellogg.northwestern.edu
Eran Shmaya: e-shmaya@kellogg.northwestern.edu

We are grateful to Lance Fortnow, Sergiu Hart, Ehud Lehrer, and Rakesh Vohra for very helpful comments.

DOI: 10.3982/TE992

each period contingent on all possible histories before that period, and the test makes a decision based on these predictions and the realized sequence of outcomes. In this general setup, without imposing other requirements, there exist tests that accept the true expert but reject strategic false experts (Dekel and Feinberg 2006, Shmaya 2008, Olszewski and Sandroni 2009a).

However, this result does not hold if some natural requirements are imposed on the test. One such requirement is that a decision to reject the expert can be made only at a finite time. Another requirement is to ask the expert to provide prediction only along the actual sequence of outcomes without giving counterfactual predictions. In the literature, tests that satisfy the first requirement are referred to as rejection tests (Olszewski and Sandroni 2009a); and the second requirement is called prequentiality (Dawid 1985).[1] The general finding (Olszewski and Sandroni 2008, Shmaya 2008) is that, against any rejection test satisfying prequentiality, a false expert has a strategy that allows him to pass it with a high probability against every realization of outcomes. The literature refers to such tests as *manipulable*, meaning that they are vulnerable to manipulation of predictions from a false expert.

Here we propose a new property, *expressibility*, that a test should satisfy, along with the rejection-test requirement and prequentiality. As discussed in Olszewski and Sandroni (2008, Section 4), a test can be regarded as a contract between the expert and those who make use of his predictions. In the corresponding contract, the principal of the forecasting service keeps the right to charge the expert with contract violation during the entire service. To implement such a contract, there should be a finite procedure that determines whether the expert fails the test for any finite history of predictions and outcomes. However, a precise definition of expressible tests is lacking in the extant literature despite the fact that many tests in the literature are expressible.

We formalize this notion by Turing computability: a test is expressible if there is a Turing machine that implements the test. A Turing machine is a finite sequence of instructions that corresponds to an algorithm: hence, for each expressible test there is a well defined procedure that determines whether the expert has failed the test for any given finite history of predictions and realizations. Moreover, it is widely believed, according to the Church–Turing thesis, that any procedure that can be expressed in words and can be implemented mechanically corresponds to a Turing machine. Thus, any contract that can be written as finitely many instructions and has a well defined procedure to implement it corresponds to an expressible test according to our definition. Indeed, all practical tests considered in the literature are expressible, including all tests proposed in the calibration literature (Foster and Vohra 1998, Lehrer 2001, Sandroni et al. 2003).

The goal of this paper is to study manipulability of expressible tests (assuming the rejection-test requirement and prequentiality). Our main results pin down the exact complexity requirements on forecasting strategies to make expressible tests manipulable. The extant literature already gives some hints about our findings. The abstract

[1] Another property, future independence in Olszewski and Sandroni (2008), is implied by these two properties.

manipulability results cited previously already imply a negative result: any expressible test is manipulable. However, it does not give any information about the complexity of the manipulating strategies. We find some answers to this problem from the calibration literature: for each calibration test, an algorithm of the polynomial-time class has been devised to implement a forecasting strategy that manipulates the test. Nevertheless, because the class of all calibration tests is only a subclass of all expressible tests, it is not clear whether forecasting strategies of the polynomial-time class can manipulate *any* expressible test.

Alternatively, Fortnow and Vohra (2009) obtain positive results by considering the computational complexity of forecasting strategies, where tests of the polynomial-time class are constructed to be nonmanipulable against forecasting strategies with time- or space-complexity constraints. Their results suggest that some expressible tests are not subject to manipulation if the expert has limited computational power, but the exact requirement on such power has not been pinned down for expressible tests. A natural starting point is to consider the set of *computable* forecasting strategies, that is, strategies that can be implemented with Turing machines without restrictions on computation resources or time.

We answer this issue with two results. In the first result, we devise an expressible test that is not manipulable by any computable forecasting strategies; in the second result, we show that if the expert has more computational power than Turing computability to implement forecasting strategies, then any expressible test is manipulable. The first result overturns the negative results in the calibration literature and shows that some expressible tests are immune to manipulability against computable forecasting strategies. In fact, we show that there are such tests that can be implemented with algorithms of the polynomial-time class. Our result then generalizes those in Fortnow and Vohra (2009) in that we consider a larger class of forecasting strategies.[2] This result is also related to another strand of literature (Olszewski and Sandroni 2009b, Al-Najjar et al. 2010), which obtains nonmanipulability results by imposing restrictions on the class of data-generating processes: a true expert passes the test only for processes in that class and the false expert fails on some process in that class. In our setup, a true expert always passes the test as long as the conditional distributions of the underlying processes admit rational probability values (but the process may not be computable) and the false expert is rejected on some computable process.

We now turn to our second result. Here we take the view that the false expert, when implementing the forecasting strategies, does not have to be constrained by Turing computability, while the test, which has to be written down as a contract, has to be expressible. Then we look for the exact complexity class of forecasting strategies against which the nonmanipulability result holds for expressible tests. To this end, we use oracle machines to model the complexity of forecasting strategies, and we classify forecasting strategies according to the arithmetic hierarchy $\{\Delta_n^0\}_{n=1}^{\infty}$. The lowest class, Δ_1^0, consists of all computable strategies, while the next class, Δ_2^0, consists of strategies that can be implemented by an oracle machine with the halting problem as the oracle. Our second result states that, for any expressible test, there exists a forecasting strategy of class

[2] In fact, as in Fortnow and Vohra (2009), our test can be modified to run in linear time.

Δ_2^0 that manipulates it. Therefore, within the arithmetic hierarchy, expressible tests can avoid the manipulation result only against forecasting strategies in the lowest class.

Those two results give the exact computational power requirement on the expert for expressible tests to be nonmanipulable: it corresponds to the class of computable strategies. This shows that expressible tests are more powerful than what one may infer from the previous literature: there are expressible tests that can handle Turing-computable forecasting strategies. It also shows that Turing-computability captures the full strength of expressible tests—false experts with computational power that is one layer higher than computability within the arithmetic hierarchy can manipulate any expressible test.

2. Tests and forecasting strategies

At each period n, an outcome s_n from a finite set S is observed. Before the observation, the principal asks the expert to deliver a probabilistic prediction, $p_n \in \Delta(S)$. The expert may use the partial history $\sigma = (p_0, s_0, \ldots, p_{n-1}, s_{n-1})$ of predictions and outcomes before period n to determine his prediction. The set of all such partial histories is given by $(\Delta(S) \times S)^{<\mathbb{N}} = \bigcup_{n=0}^{\infty}(\Delta(S) \times S)^n$, where $(\Delta(S) \times S)^0 = \{e\}$ and e is the empty sequence. Similarly, elements in $S^{<\mathbb{N}} = \bigcup_{n=0}^{\infty} S^n$ are called *partial realizations*.

A *forecasting strategy* is then a function

$$f : (\Delta(S) \times S)^{<\mathbb{N}} \to \Delta(\Delta(S)),$$

with the interpretation that $f(p_0, s_0, \ldots, p_{n-1}, s_{n-1})$ is the distribution according to which the expert randomizes his prediction at period n, where p_k and s_k are the prediction and outcome of period k for $0 \leq k < n$. The contract between the expert and the principal is written as a *test*, which is a Borel subset T of $(\Delta(S) \times S)^{<\mathbb{N}}$.[3] The expert fails the test T if $(p_0, s_0, \ldots, p_n, s_n) \in T$ for some period n.

The data-generating process is governed by an S-valued stochastic process $\mathcal{X} = (X_0, X_1, \ldots)$. Given the process $\mathcal{X}$, a forecasting strategy $f : (\Delta(S) \times S)^{<\mathbb{N}} \to \Delta(\Delta(S))$, and a test $T \subseteq (\Delta(S) \times S)^{<\mathbb{N}}$, we can compute the probability that the expert fails T over $\mathcal{X}$. That probability, denoted by $R(T, f, \mathcal{X})$, is given by

$$R(T, f, \mathcal{X}) = \mathbb{P}((P_0, X_0, \ldots, P_n, X_n) \in T \text{ for some } n),$$

where P_n, the *predictions generated by f over $\mathcal{X}$*, are $\Delta(S)$-valued random variables such that the conditional distribution of P_n given $P_0, \ldots, P_{n-1}$ and $\mathcal{X}$ is $f(P_0, X_0, \ldots, P_{n-1}, X_{n-1})$.

A natural requirement on a test is not to fail the true expert, who knows the data-generating process and makes predictions accordingly, with high probability. For an S-valued stochastic process $\mathcal{X} = (X_0, X_1, \ldots)$, let $f_{\mathcal{X}}$ be the forecasting strategy that predicts according to $\mathcal{X}$, i.e., $f_{\mathcal{X}}(p_0, s_0, \ldots, p_{n-1}, s_{n-1})$ is the dirac atomic distribution on

[3] Our definition of test is more restrictive than what is used in some other papers because it implicitly assumes the rejection-test property and prequentiality. More general definitions allow the outcome of the test to depend on the infinite sequence of predictions and outcomes or on counterfactual predictions (that is, predictions conditional on unrealized histories).

the element $p_{\mathcal{X},x} \in \Delta(S)$ that represents the conditional distribution of the process given the partial realization $x = (s_0, \dots, s_{n-1})$, i.e.,

$$p_{\mathcal{X},x}[s] = \mathbb{P}(X_n = s | X_0 = s_0, \dots, X_{n-1} = s_{n-1}). \tag{1}$$

Then $R(T, f_{\mathcal{X}}, \mathcal{X})$ is the probability that T rejects the true expert when the truth is $\mathcal{X}$. We say that a test *does not reject truth with probability* $1 - \epsilon$ if $R(T, f_{\mathcal{X}}, \mathcal{X}) < \epsilon$ for every stochastic process $\mathcal{X}$.

Here we give an example of a test that does not reject truth with probability $1 - \epsilon$.

Example 1 (Passing the true expert). Let $S = \{0, 1\}$, so that elements of $\Delta(S)$ can be identified with elements p of $[0, 1]$, where p is the probability for the outcome 1. Let $T_{N,\alpha}$ be the test that rejects the expert on all histories $(p_0, s_0, \dots, p_{n-1}, s_{n-1}) \in (\Delta(S) \times S)^{<\mathbb{N}}$ such that $n > N$ and

$$\frac{1}{n} \cdot \left(\sum_{k<n,\ p_k<1/2} (-1)^{s_k+1} + \sum_{k<n,\ p_k \geq 1/2} (-1)^{s_k} \right) > \alpha$$

for some parameters $N \in \mathbb{N}$ and $\alpha > 0$. ♢

The test $T_{N,\alpha}$ works as follows. The expert gets a penalty point whenever the outcome is far from his prediction (i.e., when $p_k < 1/2$ but $s_k = 1$ or $p_k \geq 1/2$ but $s_k = 0$) and gets credit otherwise. The test $T_{N,\alpha}$ rejects the expert if, in the long run, the penalty points exceed the credit points by a specific amount (normalized by the number of periods). The parameter α determines how strict the test is. For any α, if N is large enough, then the test $T_{N,\alpha}$ does not reject truth with probability $1 - \epsilon$.

The test $T_{N,\alpha}$ is manipulable: a forecasting strategy f ϵ*-manipulates* a test T if a false expert, by implementing f, can ignorantly pass the test T with probability greater than $1 - \epsilon$ regardless of the true data-generating process, that is, $R(T, f, \mathcal{X}) < \epsilon$ for every stochastic process $\mathcal{X}$. We give an example of a forecasting strategy that manipulates $T_{N,\alpha}$.

Example 2 (Manipulating strategy). Let $S = \{0, 1\}$ as in Example 1 and let f be the strategy given by

$$f(p_0, s_0, \dots, p_{n-1}, s_{n-1}) = \begin{cases} \dfrac{R_0}{R_0 + R_1}\delta_{1/4} + \dfrac{R_1}{R_0 + R_1}\delta_{3/4} & \text{if } R_0 > 0 \\ \delta_{3/4} & \text{otherwise;} \end{cases}$$

where

$$R_0 = \max\big\{|\{k < n | p_k \geq 1/2, s_k = 0\}| - |\{k < n | p_k \geq 1/2, s_k = 1\}|, 0\big\}$$
$$R_1 = \max\big\{|\{k < n | p_k < 1/2, s_k = 1\}| - |\{k < n | p_k < 1/2, s_k = 0\}|, 0\big\},$$

and δ_p is the dirac atomic distribution on p. For every $\alpha > 0$ and $\epsilon > 0$, the strategy f ϵ-manipulates the test $T_{N,\alpha}$ in Example 1 for sufficiently large N, that is, $R(T_{N,\alpha}, f, \mathcal{X}) < \epsilon$ for every stochastic process $\mathcal{X}$. ♢

Remark 1. The fact that the strategy in Example 2 manipulates the test follows from the standard no-regret argument using Blackwell's approachability theorem. See Lehrer (2003) for a substantially more general result using that argument.

It is desirable for the test not to be manipulable by a false expert. However, the previous literature, which studies a more general class of tests, shows that if it is a rejection test and if it does not use counterfactual predictions, then it is manipulable. Because our definition of tests already includes these two assumptions, we have the following manipulability result, which follows from Olszewski and Sandroni (2008) and Shmaya (2008).

Proposition 1. *Every test that does not reject truth with probability* $1-\epsilon$ *is* $\epsilon+\delta$*-manipulable for every* $\delta>0$.

3. Expressible tests

Here we propose another requirement for tests: the contract (written as a test) between the principal and the expert should be implementable with an algorithm. We call such tests *expressible tests* and formalize this notion by Turing computability. In Section 4, we study manipulability of expressible tests. We begin this section with some preliminaries on computability and then formulate expressible tests formally.

3.1 *Preliminaries on computability*

A function f with natural arguments and values is *computable* if there exists an algorithm that *computes* f, i.e., if n is in the domain of f, then the algorithm halts on input n and produces output $f(n)$, and if n is not in the domain of f, then the algorithm does not halt on input n and runs forever. An algorithm corresponds to a computer program in any programming language (say, the language C), running on a machine without memory or time restrictions. The formal definition is based on a model of computations using *Turing machines* (see Odifreddi 1989 for details). The celebrated Church–Turing hypothesis states that Turing computability captures our intuition of a finite procedure or an algorithm, that is, a function can be computed by an algorithm if and only if it can be computed by a Turing machine.

Here we make two remarks on computable functions. First, the domain of a computable function can be a strict subset of $\mathbb{N}$. We sometimes write $f :\subset \mathbb{N} \to \mathbb{N}$ to emphasize that the domain of f is a subset of $\mathbb{N}$. This is because an algorithm may run into an infinite loop and never produce an output for some inputs. When the domain is $\mathbb{N}$, i.e., when $f(n)$ is defined for every natural number n, we say that f is *total*. Second, the notion of computability can be extended to functions with several variables, corresponding to computer programs that get several natural numbers as input.

In fact, the generalization goes further. Many sets of mathematical objects, such as $\mathbb{N}^k$, $\mathbb{Q}$, and $\mathbb{N}^{<\mathbb{N}}$, can be *effectively identified* with $\mathbb{N}$. There are ways to *encode* elements of these sets as natural numbers, i.e., there exist computable one-to-one correspondences, called codings, between these sets and the set $\mathbb{N}$. Consider the set $\mathbb{N}^2$

as an example. Every pair (m, n) of natural numbers can be encoded as the number $(n+m)(n+m+1)/2+n$, which is a computable function. Similarly, every rational number can be encoded as a pair of natural numbers, and therefore, as a natural number. Given the coding, we can speak about computable functions to and from these sets. In what follows, we apply the notion of computability to any set Z that can be effectively identified with $\mathbb{N}$, assuming a fixed coding but without constructing the specific codes (which can be found in Odifreddi 1989).

The notion of computability can be applied to subsets of natural numbers as well. Let Z be a set that can be effectively identified with $\mathbb{N}$ and let A be a subset of Z. We say that A is computable, or *decidable*, if its characteristic function χ_A is computable. By the Church–Turing thesis, a set A is decidable if and only if there is an algorithm that determines the membership of the set A. Later we formulate a test as a subset of a set that is effectively identified with $\mathbb{N}$ and formalize the notion of expressibility by decidability.

Most sets and functions that one comes across are computable. Take, for example, the function $r : \mathbb{N}^2 \to \mathbb{N}$ such that $r(n, m) = 1$ if m divides n and $r(n, m) = 0$ otherwise. Long division gives an algorithm for this function. A more complicated example is the set of prime numbers. A number n is prime if and only if there exists no $1 < m < n$ such that m divides n. The definition already gives rise to an algorithm to check whether a number is prime: go over all those numbers m and check whether m divides n, i.e., for each such m check whether $r(n, m) = 1$; this is a computable operation because r is computable. In what follows, we take the computability of a function for granted when our definition of the function gives rise to an algorithm that computes the function.

The two examples above exhibit an important property of computability: if f is computable and g can be computed by an algorithm that *calls* f at some points, then g is also computable. In the above algorithm, checking whether a number is prime calls to the algorithm that computes divisibility. Indeed, most programming languages support such a construction by allowing the programmer to call existing functions to define new ones. We return to this point later when we define oracle computation.

The existence of uncomputable functions can be easily shown by a counting argument. Because the set of computable functions, which has the same cardinality as the set of computer programs (which are, after all, finite sequences of symbols), is countable, most functions are not computable. Related to this counting argument is the celebrated result in computability theory, the enumeration theorem (Odifreddi 1989, Theorem II.1.5), which is used in our proofs. That theorem states the existence of a binary computable function $U :\subset \mathbb{N}^2 \to \mathbb{N}$ such that, for every computable function f, there is an m such that $f = U(m, \cdot)$, i.e., such that $f(n)$ is defined if and only if $U(m, n)$ is defined, and if either is defined, we have $f(n) = U(m, n)$.

The main insight from the enumeration theorem is that there exists an effective encoding of all computer programs, say m being the code for the program p_m, and the function U corresponds to an algorithm that takes the codes as input and simulates the behavior of each program p_m when given input m. Indeed, the development of operating systems is based on this insight that the master program can manage all application programs (cf. Davis 2001). The function U is called a *universal* machine and the sequence $\phi_0 = U(0, \cdot)$, $\phi_1 = U(1, \cdot), \ldots$ is called a *computable enumeration* of computable functions. The enumeration theorem also gives rise to an example of a undecidable set: the set $H = \mathrm{dom}(U)$, which consists of all pairs (m, n) such that $U(m, n)$ is defined (i.e., such that the program p_m halts on input n), is undecidable. The set H is usually called the *halting problem*. In a more colloquial language, the halting problem cannot be solved by a Turing machine.

Finally, we introduce the notion of a *oracle machine*, which is a Turing machine that has access to a black box, called an *oracle*. In the previous example of the program that decides whether a number is a prime number, the program calls to the function r, which acts as an oracle in the sense that whenever the program calls to r, it returns values that are used by the program but are not directly computed by the program. Oracle machines generalize this idea and allow the oracle to be the characteristic function of an arbitrary subset of natural numbers (see Odifreddi 1989 for details). As such, each oracle can be identified with a subset of natural numbers. If a function f can be computed with an oracle machine with set A as the oracle, then we say that the function f is computable from A. If A is decidable, then the set of functions computable from A is the set of Turing-computable functions. However, the function χ_H, as well as any Turing-computable function, is computable from H. In fact, many other uncomputable functions are computable from H. Nevertheless, because an oracle machine consists only of finitely many instructions, the set of functions computable from a fixed oracle is still a countable set.

In the literature, there are various ways to classify the oracles, or, equivalently, subsets of natural numbers, into different complexity classes. Here we adopt the *arithmetical hierarchy*, described by Δ_n^0 sets (see Odifreddi 1989, Section IV). This classification is based on the complexity structure of quantifiers necessary to describe each subset of natural numbers as a predicate. This hierarchy respects the strength of oracles in terms of functions computable from them; more precisely, if f is computable from A for some $A \in \Delta_n^0$, then f is computable from any set $B \in \Delta_{n+1}^0$. The set Δ_1^0 consists of all decidable sets. A set $A \in \Delta_2^0$ if and only if χ_A is computable from the halting problem H (Odifreddi 1989, Proposition IV.1.16). In this sense, the halting problem is among the least uncomputable functions.

3.2 *Expressible tests*

Here we formalize the notion of expressible tests via computability. A test is expressible if and only if it can be implemented by a Turing machine or, by the Church–Turing thesis, if and only if there is an algorithm to implement the test. This definition, however, requires modifications of previous definitions of tests and forecasting strategies, because computability applies only to functions over natural numbers or subsets of natural numbers, while tests, as defined in Section 2, involve real numbers from the expert's predictions.

For this modification, we assume that the predictions made by the experts are always in rational numbers. We can relax this restriction to allow computable real numbers (which may be more natural in our context), but that requires more notation and technical details, but does keep the results intact. Because we restrict the expert to use only rational probability values and because we require the true expert to report the true data-generating process, we also restrict our attention to data-generating processes that involve only rational probability values. To discuss the *complexity* of forecasting strategies, we require the false expert's randomization over his predictions to involve only rational numbers.

To formalize these modifications, we define *admissible distributions* over an abstract finite or countable set Z as follows: a distribution μ over Z is admissible if and only if $\mu[z] \in \mathbb{Q}$ for every $z \in Z$ and $\mu[z] = 0$ for all but finitely many z's. The set of all admissible distributions over Z is denoted by $\Delta_a(Z)$. Notice that if Z can be effectively identified with $\mathbb{N}$ or if Z is finite, then $\Delta_a(Z)$ can be effectively identified with $\mathbb{N}$ as well, since $\mathbb{Q}$ can be effectively identified with $\mathbb{N}$ and finite sequences of $\mathbb{N}$ can also be effectively identified with $\mathbb{N}$.

Now we are ready to modify the definition of tests and forecasting strategies. Assuming that the experts can only make predictions in rational numbers, we define an *admissible test* as a subset T of $(\Delta_a(S) \times S)^{<\mathbb{N}}$. Notice that the set $(\Delta_a(S) \times S)^{<\mathbb{N}}$ can be effectively identified with $\mathbb{N}$ and hence T can be regarded as a subset of $\mathbb{N}$. As a result, it is then legitimate to ask whether an admissible test T is decidable. Similarly, define an *admissible forecasting strategy* to be a function $f : (\Delta_a(S) \times S)^{<\mathbb{N}} \to \Delta_a(\Delta_a(S))$, with the interpretation that $f(p_0, s_0, \ldots, p_{n-1}, s_{n-1})$ is the distribution according to which the expert randomizes the prediction at period n, given the previous predictions p_k and outcomes s_k, $k \leq n-1$. Notice that, again, both $(\Delta_a(S) \times S)^{<\mathbb{N}}$ and $\Delta_a(\Delta_a(S))$ can be effectively identified with $\mathbb{N}$, and hence an admissible forecasting strategy can be regarded as a function over natural numbers. Hence it is also legitimate to ask whether an admissible forecasting strategy f is computable and, for any given oracle, whether f is computable from that oracle.

Finally, we restrict the underlying data-generating process to be admissible as well (again, as mentioned earlier, our results do not change if we allow computable real numbers). We say that an S-valued stochastic process $X_0, X_1, \ldots$ is *admissible* if the conditional distribution, $p_{\mathcal{X},x}$, is an admissible distribution for every partial realization x, where $p_{\mathcal{X},x}$ is given by (1). Notice that if a stochastic process $\mathcal{X}$ is admissible, then $f_{\mathcal{X}}$ as defined in Section 2 is an admissible forecasting strategy.

As in Section 2, we consider only tests that do not reject true experts, but restrict our attention to admissible data-generating processes only: we say that a test T *does not reject admissible truth with probability* $1-\epsilon$ if $R(T, f_{\mathcal{X}}, \mathcal{X}) < \epsilon$ for every admissible stochastic process $\mathcal{X}$. Here we assume that the true expert always makes predictions according to the true data-generating process, even if it is not computable. This assumption is consistent with the previous literature. However, it implies that the true expert's forecasting is not bounded by complexity constraints, while the false expert may be. Alternatively, we could require the test to pass computable truth only. Both of our results still hold under this alternative assumption.

Given these modified definitions, we can then speak of expressible tests and complexity of forecasting strategies in terms of the arithmetic hierarchy. First we define expressible tests.

Definition 1. An admissible test $T \subset (\Delta_a(S) \times S)^{<\mathbb{N}}$ is *expressible* if the set T is decidable.

All tests considered in the calibration literature are expressible, including the test $T_{N,\alpha}$ in Example 1. It is immediate from Proposition 1 that expressible tests are manipulable. We are interested, however, in identifying the complexity classes of the forecasting strategies for which the manipulability result does or does not hold. The complexity of an admissible forecasting strategy f is measured by the strength of the oracle necessary to implement f with an oracle machine, and we employ the arithmetic hierarchy $\{\Delta_n^0\}_{n=1}^{\infty}$ to classify the strengths of oracles. As mentioned earlier, the class of computable admissible forecasting strategies correspond to the bottom hierarchy Δ_1^0, and the class of admissible strategies that are computable from the halting problem H corresponds to the second lowest class Δ_2^0. It turns out, as we show in the next section, that, for expressible tests, these two are the relevant complexity classes for the manipulability result to hold or not to hold.

4. Results

We begin with the nonmanipulability result. Theorem 1 states that there exists an expressible test such that for any computable forecasting strategy f, the test rejects the false expert, who knows nothing about the underlying data-generating process, with high probability for some data-generating process. Moreover, it states that this is true for a computable data-generating process, where an admissible stochastic process $\mathcal{X} = X_0, X_1, \ldots$ is said to be *computable* if $f_{\mathcal{X}}$ is computable. In Theorem 1, we require the false expert to have access only to computable forecasting strategies and, correspondingly, we show the nonmanipulable result against computable stochastic processes. Notice, however, that the test passes a true expert for all admissible stochastic processes, not only computable ones.

Theorem 1. *For every $\epsilon > 0$, there exists an expressible test T that does not reject admissible truth with probability $1 - \epsilon$ and such that for every computable strategy f, there exists some computable S-valued stochastic process $\mathcal{X}$ for which $R(T, f, \mathcal{X}) > 1 - \epsilon$.*

Theorem 1 is similar to the results in Fortnow and Vohra (2009), with the difference that the constructed tests there are all within the polynomial-time complexity class. However, as Remark 4 shows, any expressible test can be modified to an equivalent test (in the sense that against any realization, one test rejects the expert if and only if the other test rejects) that is within the polynomial-time complexity class. Therefore, Theorem 1 generalizes those results in the sense that it broadens the class of forecasting strategies against which the nonmanipulability result holds for expressible tests. The formal proof of Theorem 1 is given in Section 5, but we give an outline of it next.

Sketch of the proof of Theorem 1. The proof relies on a simple observation of Dawid (1985): for a given countable set $\mathcal{F} = \{f_1, f_2, \ldots\}$ of forecasting strategies, a test that does not reject truth with probability $1-\epsilon$ can be devised such that, for each strategy f_m, there is a partial realization x^m against which f_m is rejected. This is especially easy to see if all strategies in $\mathcal{F}$ are deterministic: for f_m, consider a deterministic sequence $x^m = (s_0^*, s_1^*, \dots, s_n^*)$ such that s_k^* is the least likely outcome (hence has probability no greater than $1/2$) conditional on $(s_0^*, \dots, s_{k-1}^*)$ according to f_m (i.e., according to the data-generating process $\mathcal{X}$ such that $f_{\mathcal{X}} = f_m$), and take the test T_m that rejects the expert under x^m and the predictions made by f_m along x^m. Moreover, we can choose the length n of the partial realization x^m to be so large that T_m rejects the truth with probability $\epsilon/2^m$. Let $T = \bigcup_m T_m$ be the test that rejects the expert if he fails in any of the tests T_m's. Then T rejects each f_m against x^m and still does not reject truth with probability $1-\epsilon$.

Applying this technique to our theorem, given that the set of computable forecasting strategies is a countable set, shows that (see the proof for accommodating randomization in the above argument) there exists a nonmanipulable test for computable strategies. However, we need to make sure that such a test is expressible. To this end, we first show that the construction of the tests T_m involves only computable operations. This is done in Lemma 1. However, this result does not guarantee that the ultimate test $T = \bigcup_m T_m$ is decidable. The crucial observation that allows us to construct a decidable T is the fact that we can enumerate all computable strategies in a computable manner, according to the enumeration theorem mentioned in Section 3.1. □

As made clear in the above argument, it is the enumeration theorem that ultimately ties the computability of the strategies available to the false expert to the computability of the test we construct. Indeed, Theorem 1 would still hold if we consider tests that are decidable relative to a fixed oracle A and the set of forecasting strategies that are computable relative to the same oracle A. In particular, if the false expert is restricted to forecasting strategies from the class Δ_n^0, then there is a test in the class of Δ_n^0 that is immune to manipulability for any $n \geq 1$.

Theorem 1 shows that expressible tests can be used to deal with computable forecasting strategies, but it remains silent about how broad the class of forecasting strategies can be for those tests to be immune to manipulations. Our second result, a manipulability result, shows that all expressible tests are manipulable if we allow the expert to use forecasting strategies that are more complicated than computable ones. More precisely, we show that for any expressible test, there is a forecasting strategy computable from the halting problem H that passes the test with high probability against any underlying data-generating process; those strategies lie in the next arithmetic hierarchy to the computable strategies.

Theorem 2. *Let T be an expressible test that does not reject admissible truth with probability $1-\epsilon$. Then, for every $\delta > 0$, there exists an admissible forecasting strategy f computable from the halting problem H that $(\delta+\epsilon)$-manipulates T.*

The proof of Theorem 2 is in Section 5, and, as with the previous theorem, we give a sketch of the proof here.

Sketch of proof of Theorem 2. The proof is based on tracking the computability restrictions in the proof of Olszewski and Sandroni (2008).

Consider first the case of a *finite* test T that depends only on predictions and outcomes made in a bounded horizon $n \in \mathbb{N}$, i.e., if the expert does not fail in the first n days, then he is off the hook. In this case, Sandroni (2003) proves that the test is manipulable. Moreover, standard arguments that are used in the literature show that there is a manipulating strategy that confines the expert's forecast at every stage to a finite grid of $\Delta(S)$, and it is also easy to extend the result so that the randomization employed by the expert over elements of this grid is confined to probability values in the same grid. We establish these results formally in Lemma 2.

Thus there is a finite set of strategies such that at least one of them manipulates the test. Since checking whether a strategy manipulates the test is a computable operation (it involves going over all realizations of length n, and for each of them, checking that the probability of failing is sufficiently small), it follows that every finite test has a computable manipulating strategy.

Now consider any (potentially infinite) admissible test T. The false expert seeks a strategy that guarantees that the chance of eventual failure is small. Equivalently, the false expert seeks a strategy that has a small chance of failure in any of the finite tests T_n, where T_n, the n-periods restriction of T, is the finite test in which the expert fails if he fails T before period n and passes otherwise. Thus the expert faces a sequence $T_0, T_1, T_2, \ldots$ of tests of increasing difficulty and he seeks a strategy that passes all of them simultaneously. Proving the existence of such a strategy calls for some compactness argument (like the one used by Olszewski and Sandroni 2008 in their appeal to Fan's theorem). This is also where computability breaks down: at day 0, the expert is already required an infinite foresight—he has to plan his prediction for that day to make it compatible with a good manipulating strategy for all tests T_n. Checking whether a certain plan fulfills this requirement requires an appeal to the halting problem as an oracle. To show that there exists a manipulating strategy that is computable relative to the halting problem, we reduce that existence problem to a problem of finding an infinite branch in a finitely splitting tree and use the Kreisel basis lemma (Odifreddi 1989, Proposition V.5.31 and Proposition IV.1.16), the computable version of the Konig lemma, which states that if a finitely splitting tree is infinite, then there is an infinite branch that is computable relative to the halting problem.

The finitely splitting tree is constructed as follows: for each finite horizon n, the forecasting strategy can be written as a sequence of functions $f^n = (f_0, \ldots, f_{n-1})$, where f_k maps partial realizations of length k to a mixture of predictions. As mentioned earlier, we can let f_k be a distribution over a finite grid of predictions that use probability values over a finite grid, and this makes the set of possible f_k's a finite set. The tree $\mathcal{T}$ contains f^n's that manipulate the test T_n. If $\mathcal{T}$ has an infinite branch, which corresponds to an infinite sequence $f = (f_0, f_1, \ldots, f_n, \ldots)$, then f^n manipulates T_n for each n and hence f manipulates T. The tree $\mathcal{T}$ is a decidable set because checking whether a finite strategy manipulates a finite test is a computable operation; $\mathcal{T}$ is infinite because there is a manipulating strategy for each n. Applying the Kreisel basis lemma, we obtain a strategy that manipulates T and is computable relative to the halting problem. □

Remark 2. Theorem 2 still holds if we require the test to pass a true expert only for computable stochastic processes. This follows from the fact that the proof of Theorem 2 needs only the tests T_n to pass a true expert.

Remark 3. As with Theorem 1, Theorem 2 can be extended to computability relative to a fixed oracle. For any test that is decidable relative to an oracle A, there is a manipulating strategy that is computable relative to A', where A' solves the halting problem relative to A. As a result, if the test belongs to the class Δ^0_n, then there is a manipulating strategy that belongs to the class Δ^0_{n+1} for any $n \geq 1$.

5. Proofs

5.1 *Proof of Theorem 1*

Notation. For every $q = (p_0, \dots, p_{n-1}) \in \Delta_a(S)^n$ and $x = (s_0, \dots, s_{n-1}) \in S^n$, let

$$\rho(q, x) = \prod_{k=0}^{n-1} p_k[s_k] \tag{2}$$

be the probability that Nature produces outcomes x if she randomizes according to the predictions q. In addition, for every admissible strategy $f : (\Delta_a(S) \times S)^{<\mathbb{N}} \to \Delta_a(\Delta_a(S))$, let

$$\pi(f, x, q) = \prod_{k=0}^{n-1} f(p_0, s_0, \dots, p_{k-1}, s_{k-1})[p_k] \tag{3}$$

be the probability that an expert who uses f produces predictions q along the partial realization x. Both ρ and π are computable functions.

For every partial realization $x^* = (s^*_0, \dots, s^*_{n-1}) \in S^n$ and every rational $\epsilon > 0$, let $T(x^*, \epsilon)$ be the test that is given by the decidable set of all sequences $\sigma = (p_0, s_0, \dots, p_{n-1}, s_{n-1}) \in (\Delta_a(S) \times S)^n$ of predictions and outcomes such that $s_k = s^*_k$ for every $0 \leq k < n$ and $\rho(q, x^*) < \epsilon$ for $q = (p_0, \dots, p_{n-1})$. Thus, under $T(x^*, \epsilon)$, the expert is rejected if the partial realization x^* occurs and he gave this realization a low probability. The test $T(x^*, \epsilon)$ then passes the true expert with probability $1 - \epsilon$.

By the enumeration theorem, there exists a computable enumeration, $\{\varphi_m\}_{m \in \mathbb{N}}$, of all computable functions $\varphi_m :\subset (\Delta_a(S) \times S)^{<\mathbb{N}} \to \Delta_a(\Delta_a(S))$. Moreover, the function $U(m, \sigma) = \varphi_m(\sigma)$ (equality here means that the values are equal when φ_m is defined and both functions are undefined otherwise) is computable.

Lemma 1. *Let $\epsilon > 0$ be a fixed rational number. There exists a computable function $\theta :\subset \mathbb{N} \to S^{<\mathbb{N}}$ such that, for all natural numbers m, if φ_m is a strategy (i.e., a total function), then $m \in \mathrm{dom}(\theta)$ and there exists some computable S-valued stochastic process $\mathcal{X}$ such that $R(T(\theta(m), \epsilon/2^{m+1}), \varphi_m, \mathcal{X}) > 1 - \epsilon$, i.e., the strategy φ_m fails the test $T(\theta(m), \epsilon/2^{m+1})$ with probability at least $1 - \epsilon$ over $\mathcal{X}$.*

Proof. Let $m \in \mathbb{N}$. We describe how to compute $\theta(m)$. Let k be a natural number such that $\epsilon > 1/2^k$ and let $n = m + 2k + 1$.

Let $\mu \in \Delta_a((\Delta_a(S) \times S)^n)$ be given by

$$\mu[q, x] = \pi(\varphi_m, x, q) \cdot \rho(q, x)$$

for every $q = (p_0, \dots, p_{n-1}) \in (\Delta_a(S))^n$ and $x = (s_0, \dots, s_{n-1}) \in S^n$, where ρ and π are given by (2) and (3). Note that if $f = \varphi_m$ is total, then the computation of μ halts on m, and in this case μ is the stochastic process $P_0, X_0, \dots, P_{n-1}, X_{n-1}$ of predictions P_k created by f and outcomes X_k randomized by Nature according to these predictions.

Let μ_X be the marginal distribution of μ over S^n and choose $x^* = (s^*_0, \dots, s^*_{n-1})$ such that

$$\mu_X[x^*] = \sum_{q \in \Delta_a(S)^n} \mu[q, x^*] \leq 1/2^n < \epsilon^2/2^{m+1}. \tag{4}$$

Then, in particular, it follows that if $\mathcal{X} = X_0, X_1, \dots$ is the admissible S-valued stochastic process such that $X_k = s^*_k$ for every $0 \leq k < n$ and $X_n, X_{n+1}, \dots$ are independent and identically distributed uniformly over S, then

$$R(T(x^*, \epsilon), f, \mathcal{X}) = \sum_{q | \rho(q, x^*) < \epsilon/2^{m+1}} \pi(f, x^*, q) > 1 - \epsilon,$$

where the inequality follows from (4) with $f = \varphi_m$. □

Proof of Theorem 1. Let T be given by $T = \bigcup_{m \geq 0} T(\theta(m), \epsilon/2^{m+1})$, where θ is the computable function from Lemma 1 and $T(\theta(m), \epsilon/2^{m+1}) = \varnothing$ if $\theta(m)$ is undefined, so that the expert fails T if and only if he fails one of the tests $T(\theta(m), \epsilon/2^{m+1})$. Since the test $T(\theta(m), \epsilon/2^{m+1})$ fails the truth with probability less than $\epsilon/2^{m+1}$, it follows that T fails the truth with probability less than $\sum_m \epsilon/2^{m+1} = \epsilon$. Finally, for every computable strategy f, if m is such that $f = \varphi_m$, then by Lemma 1, f fails $T(\theta(m), \epsilon/2^{m+1})$ (hence also T) with probability at least $1 - \epsilon$ over some computable S-valued stochastic process $\mathcal{X}$.

The test T, however, may not be decidable. Here we modify it to an equivalent test that is decidable. Let A be the algorithm that computes the function $\theta(m)$. First we construct a decidable set $\tilde{T} \subseteq (\Delta_a(S) \times S)^{<\mathbb{N}} \times \mathbb{N} \times \mathbb{N}$ as follows. The triplet $(\sigma, m, n) \in \tilde{T}$ if and only if the following conditions hold: (a) the algorithm A halts and gives output $\theta(m)$ at input m within n steps; (b) $\sigma \in T(\theta(m), \epsilon/2^{m+1})$. The set $\tilde{T}$ is decidable because it checks only whether A halts within a fixed number of steps instead of checking whether it eventually halts. Then, for all $\sigma \in (\Delta_a(S) \times S)^{<\mathbb{N}}$, $\sigma \in T$ if and only if $(\sigma, m, n) \in \tilde{T}$ for some $(m, n) \in \mathbb{N}^2$.

Now let $T^* \subseteq (\Delta_a(S) \times S)^{<\mathbb{N}}$ be the test such that $\sigma = (p_0, s_0, \dots, p_l, s_l) \in T^*$ if and only if $l = (n + m)(n + m + 1)/2 + n$ and $(\tau, m, n) \in \tilde{T}$ for some initial segment τ of σ.[4] Then T^* is a decidable test: to decide whether $\sigma \in T^*$ with $|\sigma| - 1 = l = (n + m)(n + m + 1)/2 + n$, the algorithm goes over all initial segments τ of σ and for every

[4] Recall that for each $l \in \mathbb{N}$, there is a unique pair (m, n) for this equality to hold; the number l is the code for the pair (m, n).

such τ, checks whether $(\tau, m, n) \in \tilde{T}$ by calling the algorithm that decides the membership of $\tilde{T}$. Moreover, $\tau \in T$ if and only if $\sigma \in T^*$ for some extension σ of τ. Therefore, T and T^* are equivalent in the sense that they reject the expert over the same infinite histories $(p_0, s_0, p_1, s_1, \ldots)$. □

Remark 4. For comparison with Fortnow and Vohra (2009), we note that the argument in the last paragraph in the proof of Theorem 1 can be modified to make the test run in polynomial time. The test T is *polynomial* if there exist a polynomial function $\rho : \mathbb{N} \to \mathbb{N}$ and an algorithm A such that, for every $\sigma = (p_0, s_0, \ldots, p_n, s_n) \in (S \times \Delta(S))^{<\mathbb{N}}$, A takes at most $\rho(n)$ steps over the input σ and decides whether $\sigma \in T$. Let T be an expressible test, that is, there is an algorithm A' that outputs 1 on σ if $\sigma \in T$ and that outputs 0 on σ otherwise. Consider the test T^* such that $\sigma = (p_0, s_0, \ldots, p_n, s_n) \in T^*$ if and only if A' outputs 1 on $(p_0, s_0, \ldots, p_k, s_k)$ after at most l steps for some $k, l \leq \sqrt{n}$. The test T^* can be implemented with the following algorithm A. Given input $\sigma = (p_0, s_0, \ldots, p_n, s_n)$, run algorithm A' on $\sigma^k = (p_0, s_0, \ldots, p_k, s_k)$ for $k \leq \sqrt{n}$ sequentially; if A' halts on some σ^k with output 1 within l steps for some $l \leq \sqrt{n}$, then A halts immediately and outputs 1; otherwise A outputs 0. This ensures that A runs in polynomial time with $\rho(n) = Cn$ for some constant C. Moreover, T and T^* are equivalent in the sense that they reject the expert over the same infinite histories $(p_0, s_0, p_1, s_1, \ldots)$.

5.2 *Proof of Theorem 2*

Fix, once and for all, a test T as in Theorem 2 and $\delta > 0$. For every n, let T_n be the *n-periods restriction of T*, i.e., the test that is given by all sequences $(p_0, s_0, \ldots, p_k, s_k) \in T$ such that $k < n$. Since $T_n \subseteq T$, it follows that T_n also does not reject admissible truth with probability $1 - \epsilon$. Since $T = \bigcup_n T_n$, it follows that

$$R(T_n, f, \mathcal{X}) \xrightarrow[n \to \infty]{} R(T, f, \mathcal{X}) \tag{5}$$

for every S-valued stochastic process $\mathcal{X} = X_0, X_1, \ldots$ and every strategy $f : (\Delta(S) \times S)^{<\mathbb{N}} \to \Delta(\Delta(S))$. We first give a lemma that shows that a manipulation strategy can be modified to have probability values over a finite grid, which in turn is a distribution over predictions over a finite grid. Let Z be a finite set. For every natural number $M > 0$, we denote by $\Delta_M(Z)$ the set of distributions μ over Z such that $\mu[z] \in \{0, 1/M, 2/M, \ldots, 1\}$ for every z.

Lemma 2. *Let $M, K : \mathbb{N} \to \mathbb{N}$ be given by $M(k) = \lceil |S| \cdot 2^k / \delta \rceil$ and $K(k) = \lceil |S|^{M(k)+1} \cdot 2^k / \delta \rceil$. Then for every n, there exists a strategy f that $\epsilon + \delta$-manipulates T_n such that $f(\sigma) \in \Delta_{K(k)}(\Delta_{M(k)}(S))$ for every k and every sequence $\sigma = (p_0, s_0, \ldots, p_{k-1}, s_{k-1})$ of past predictions and outcomes.*

To prove Lemma 2, we use three claims. The first claim states that every distribution over a finite set Z can be approximated by a distribution with probability values on a finite grid. We use the L_1 metric: for any $\mu, \nu \in \Delta(Z)$, $\|\mu - \nu\|_1 = \sum_{z \in Z} |\mu[z] - \nu[z]|$. The proof of the claim is easy and is omitted.

Claim 1. Let Z be a finite set. Then for every $\mu \in \Delta(Z)$ and every M, there exists $\mu' \in \Delta_M(Z)$ such that $\|\mu' - \mu\|_1 < |Z|/M$.

The proofs of Claims 2 and 3 below are omitted as well, since similar arguments have already appeared in previous papers (Olszewski and Sandroni 2008, Shmaya 2008, Lemma 1). Claim 2 states that if the expert gives prediction that are close enough to the truth, he still passes a test that does not reject a true expert. Claim 3 states that a slight perturbation of a strategy does not greatly change the probability of failing a particular test.

Claim 2. Let $\delta > 0$, let $\mathcal{X} = X_0, X_1, \dots$ be an S-valued stochastic process, and let f be a forecasting strategy such that, for every sequence $\sigma = (p_0, s_0, \dots, p_{k-1}, s_{k-1})$, $f(\sigma)$ is the dirac atomic distribution on an element $p \in \Delta(S)$ such that $\|p - p_{\mathcal{X},x}\|_1 < \delta/2^k$, where $x = (s_0, \dots, s_{k-1})$ and $p_{\mathcal{X},x}$ is given by (1). Then $R(T, f, \mathcal{X}) < \epsilon + \delta$.

Claim 3. Let $\delta > 0$ and let $f, \tilde{f} : (\Delta_a(S) \times S)^{<\mathbb{N}} \to \Delta(\Delta_a(S))$[5] be two strategies such that $\|f(\sigma) - \tilde{f}(\sigma)\|_1 < \delta/2^k$ for every $k \in \mathbb{N}$ and every sequence $\sigma = (p_0, s_0, \dots, p_{k-1}, s_{k-1})$ of predictions and outcomes. Then $|R(T, f, \mathcal{X}) - R(T, \tilde{f}, \mathcal{X})| < \delta$ for every S-valued stochastic process $\mathcal{X}$.

Proof of Lemma 2 (Sketch). Fix n. Then for every stochastic process $\mathcal{X}$, there exists by Claim 1 a strategy f such that, for every sequence $\sigma = (p_0, s_0, \dots, p_{k-1}, s_{k-1})$ of predictions and outcomes, $f(\sigma)$ is the dirac atomic distribution on an element $p \in \Delta_{M(k)}(S)$ such that $\|p - p_{\mathcal{X},x}\|_1 < \delta/2^k$ (notice that for Claim 1 to be applicable, $M(k)$ must satisfy $|S|/M(k) \le \delta/2^k$), where $x = (s_0, \dots, s_{k-1})$ and $p_{\mathcal{X},x}$ is given by (1). By Claim 2, it follows that $R(T_n, f, \mathcal{X}) < \epsilon + \delta$.

By a minmax argument as in Sandroni (2003), it follows from the last observation that there is a prediction strategy f that $\epsilon + \delta$-manipulates the test T_n and such that $f(\sigma) \in \Delta(\Delta_{M(k)}(S))$ for every $\sigma = (p_0, s_0, \dots, p_{k-1}, s_{k-1})$. Notice that the minmax argument works because T_n can be used to construct a zero-sum game with a finite set of strategies for each player.

By Claim 1, f can be approximated by a strategy $\tilde{f}$ such that $\tilde{f}(\sigma) \in \Delta_{K(k)}(\Delta_{M(k)}(S))$ (notice that $K(k)$ satisfies $|S|^{M(k)+1}/K(k) \le \delta/2^k$) and $|\tilde{f}(\sigma) - f(\sigma)| < \delta/2^k$ for every $\sigma = (p_0, s_0, \dots, p_{k-1}, s_{k-1})$, and by Claim 3, $\tilde{f}$ is a strategy that $(\epsilon + 2\delta)$-manipulates T_n. □

Proof of Theorem 2. Let $R_n = \Delta_{M(0)}(S) \times S \times \cdots \times \Delta_{M(n-1)}(S) \times S$ be the set of all possible sequences $\sigma = (p_0, s_0, \dots, p_{n-1}, s_{n-1})$ of the n-stage past predictions and outcomes if the expert uses the strategy f as in Lemma 2, and let $F_n = (\Delta_{K(n)}(\Delta_{M(n)}(S)))^{R_n}$ be the set of contingent mixtures in day n given past history when the kth day's mixture is restricted to $\Delta_{K(k)}(\Delta_{M(k)}(S))$. Let $\mathcal{T} \subseteq \bigcup_{n\ge 0} F_0 \times \cdots \times F_n$ be the finitely splitting tree of all elements $f^n = (f_0, \dots, f_{n-1}) \in F_0 \times \cdots \times F_{n-1}$ such that the following condition is satisfied: If g is an admissible strategy for which $g(\sigma) = f_k(\sigma)$ for every sequence $\sigma \in R_k$,

[5] Notice that here we assume that both strategies are mixtures over rational predictions, but not necessarily admissible.

then g $(\epsilon+\delta)$-manipulates T_n. We call such a strategy g an extension of f^n. Notice that f^n does not define an admissible forecasting strategy—it does not define predictions for many partial histories; however, it is also true that for any realization, one extension of f^n fails on T_n if and only if any other extension fails on T_n.

The set $\mathcal{T}$ is a tree, that is, if $f^n=(f_0,\dots,f_{n-1})\in\mathcal{T}$, then $f^k=(f_0,\dots,f_{k-1})\in\mathcal{T}$ for any $k<n$. This result follows from the fact that $T_k\subseteq T_n$ and hence if any admissible strategy g that extends f^n $(\epsilon+\delta)$-manipulates T_n, then any admissible strategy g' that extends f^k $(\epsilon+\delta)$-manipulates T_k. Note that to check whether an extension g of f^n manipulates T_n, it is sufficient to check that the probability that an expert who uses f_k to predict in day k $(k=0,\dots,n-1)$ will be rejected by day n is less than $(\epsilon+\delta)$ for every partial realization $x=(s_0,\dots,s_n)$. Since the number of such partial realizations is finite, it follows that $\mathcal{T}$ is a decidable set.

By Lemma 2, for every n, the tree $\mathcal{T}$ has a node of length n. Therefore, by the Kreisel basis lemma, $\mathcal{T}$ admits an infinite branch $(f_0,f_1,\dots)$ that is computable relative to the halting problem. If g is the corresponding admissible strategy such that $g(\sigma)=f_k(\sigma)$ for every sequence $\sigma\in R_k$ and $g(\sigma)$ is the atomic dirac measure on the uniform distribution for other σ's, then g is a strategy that $(\epsilon+\delta)$-manipulates T_n for every n. By (5), it follows that g also $(\epsilon+\delta)$-manipulates T. Also, g is computable relative to the halting problem because the infinite branch $(f_0,f_1,\dots)$ is. □

References

Al-Najjar, Nabil I., Alvaro Sandroni, Rann Smorodinsky, and Jonathan Weinstein (2010), "Testing theories with learnable and predictive representations." *Journal of Economic Theory*, 145, 2203–2217. [265]

Davis, Martin (2001), *Engines of Logic*. Norton, New York. [270]

Dawid, A. Philip (1985), "The impossibility of inductive inference." *Journal of the American Statistical Association*, 80, 340–341. [264, 272]

Dekel, Eddie and Yossi Feinberg (2006), "Non-Bayesian testing of a stochastic prediction." *Review of Economic Studies*, 73, 893–906. [264]

Fortnow, Lance and Rakesh V. Vohra (2009), "The complexity of forecast testing." *Econometrica*, 77, 93–105. [265, 272, 277]

Foster, Dean P. and Rakesh V. Vohra (1998), "Asymptotic calibration." *Biometrika*, 85, 379–390. [264]

Lehrer, Ehud (2001), "Any inspection is manipulable." *Econometrica*, 69, 1333–1347. [264]

Lehrer, Ehud (2003), "A wide range no-regret theorem." *Games and Economic Behavior*, 42, 101–115. [268]

Odifreddi, Piergiorgio (1989), *Classical Recursion Theory*, volume 125 of Studies in Logic and the Foundations of Mathematics. Elsevier, Amsterdam. [268, 269, 270, 274]

Olszewski, Wojciech and Alvaro Sandroni (2008), "Manipulability of future-independent tests." *Econometrica*, 76, 1437–1466. [264, 268, 274, 278]

Olszewski, Wojciech and Alvaro Sandroni (2009a), "A nonmanipulable test." *The Annals of Statistics*, 37, 1013–1039. [264]

Olszewski, Wojciech and Alvaro Sandroni (2009b), "Strategic manipulation of empirical tests." *Mathematics of Operations Research*, 34, 57–70. [265]

Sandroni, Alvaro (2003), "The reproducible properties of correct forecasts." *International Journal of Game Theory*, 32, 151–159. [274, 278]

Sandroni, Alvaro, Rann Smorodinsky, and Rakesh V. Vohra (2003), "Calibration with many checking rules." *Mathematics of Operations Research*, 28, 141–153. [264]

Shmaya, Eran (2008), "Many inspections are manipulable." *Theoretical Economics*, 3, 367–382. [264, 268, 278]

Submitted 2011-4-26. Final version accepted 2012-4-6. Available online 2012-4-6.