On Learning To Coordinate:

Random Bits Help, Insightful Normal Forms, and Competency Isomorphisms

John Case [a] Sanjay Jain [b] Franco Montagna [c] Giulia Simi [c] Andrea Sorbi [c]

[a] *Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716-2586 (USA). Email: case@cis.udel.edu*

[b] *School of Computing, National University of Singapore, Singapore 117543, Republic of Singapore. Email: sanjay@comp.nus.edu.sg*

[c] *Dipartimento di Scienze Matematiche ed Informatiche "R. Magari", University of Siena, 53100 Siena, Italy. Email: {montagna, simi, sorbi}@unisi.it*

Abstract

A mere bounded number of random bits judiciously employed by a probabilistically correct algorithmic coordinator is shown to increase the power of learning to coordinate compared to deterministic algorithmic coordinators. Furthermore, these probabilistic algorithmic coordinators are provably *not* characterized in power by *teams* of deterministic ones.

An insightful, enumeration technique based, normal form characterization of the classes that are learnable by total computable coordinators is given. These normal forms are for insight only since it is shown that the complexity of the normal form of a total computable coordinator can be infeasible compared to the original coordinator.

Montagna and Osherson showed that the competence class of a total coordinator cannot be strictly improved by another total coordinator. It is shown in the present paper that the competencies of any two total coordinators are the same modulo isomorphism. Furthermore, a completely effective, index set version of this competency isomorphism result is given, where all the coordinators are total computable. We also investigate the competence classes of total coordinators from the points of view of topology and descriptive set theory.

1 Introduction

The learning theory paradigms descending from Gold's [9], as summarized say in [10], do not generally take into account the delicacies of learning with

Preprint submitted to Elsevier Science *11 March 2007*

timing, feedback, and interaction as one would see with agents learning to *coordinate* their activities in real time. In practice such agents could range from the environment together with individual muscle and brain cells of animals to furniture movers verbally negotiating in real time how to get a large, irregularly shaped object through a doorway. On the other hand, also descending from the Gold paradigm, are [5,11] which study learning to win reactive process-control games and [3] which studies learning to predict concepts which are changing with time. These studies, in a sense, capture *some* aspects of learning to coordinate. A new paradigm (under Gold's) very specifically aimed at learning to coordinate was introduced in [14]. In this paradigm one studies, say, two, agents, called *players* (or *coordinators*) which simultaneously play bits (i.e., elements of $\{0,1\}$), *each taking into account the prior bits played by the other.* [1] Their goal, successful coordination, is to have their bit streams perfectly match past some point. [14] gives an example of two people who show up in a park each day at one of noon (bit 0) or 6pm (bit 1); each *silently* watches the others past behavior; and each *tries*, based on the past behavior of the other, to show up eventually exactly when the other shows up. If they manage it, they have learned to coordinate.

Mathematically, we model players as (partial) functions from (finite) bit strings to bits (or bit strings). A player can be, for example, total, partial computable, computable, or probabilistic (the latter with some probability of success).

In the present paper we extend [14] in three interesting directions which we will discuss briefly in turn. Each direction is considered in a separate section below and, in the present section, for each direction, we informally discuss only our main results in that direction.

We are interested in single coordinators which do or do not (learn to) coordinate with a whole class $\mathcal{C}$ of coordinators. For example, some algorithmic player learns to coordinate with $\mathcal{C}$ = the class of polynomial-time computable, 0-1 valued functions.

The main results of Section 3 concern probabilistic algorithmic coordinators PM surprisingly beating deterministic ones (Theorems 3, 4, 6, and 7) or beating probabilistic ones (Corollary 9) — each at coordinating with some class $\mathcal{C}$ of algorithmic deterministic players. One nice property of these main results of Section 3 is that each features a PM which tosses a bounded number of coins, i.e., the PM's do not require coin tosses on infinitely many inputs. The PM's and other (deterministic) machines which are shown to exist also do not have to remember very much about their past inputs — in memory limited senses identical to or similar to the senses studied in [16, Page 66] and [4,7,8,14].

[1] More agents than two can easily be accommodated in the model as can allowing outputs besides bits.

Theorems 3 and 4 essentially show that there are classes $\mathcal{C}$ of deterministic algorithmic coordinators such that *no* deterministic algorithmic coordinator can (learn to) coordinate with each element of $\mathcal{C}$, but *some* probabilistic algorithmic coordinator can.

For $1 \leq i \leq m$, $\mathbf{Team}^i_m\mathbf{Coord}$ denotes the collection of classes $\mathcal{C}$ of algorithmic *deterministic* players such that we have m deterministic algorithmic coordinators $M^1, \ldots, M^m$ so that for each element F of $\mathcal{C}$, at least i of $M^1, \ldots, M^m$ coordinate with F.

In the setting of learning programs for total computable functions, team learning by deterministic machines is completely characterized by single machines learning probabilistically and vice versa [17–19]. Furthermore, the analog of $\mathbf{Team}^i_m$ in *that* context has exactly the inferring power of single machines which succeed with probability at least $\frac{i}{m}$. One might suspect that such a characterization holds for learning to coordinate, and that Theorems 3 and 4 mentioned above are readily explainable as deterministic team learning in disguise. Theorems 6 and 7 show this is not the case, and that no such characterization holds in the case of learning to coordinate! A few random bits make a big difference in learning to coordinate. One wonders, then, in conceiving the brain and its environment as a collection of coordinators working together (somewhat as in [13]) for tasks such as muscle movement and speech, if random bits may need to be employed to achieve learning to coordinate.

Corollary 9 shows that, for probabilities $p < q$, there are classes $\mathcal{C}$ of deterministic algorithmic coordinators, such that some probabilistic algorithmic coordinator PM learns to coordinate with each player in $\mathcal{C}$ with probability p, but none can do it with probability q.

The PM's of Theorems 3 and 6 and Corollary 9 are additionally *blind*, i.e., they depend functionally only on the *length* of the bit strings they see. Theorems 4 and 7 are uniformizations of Theorems 3 and 6, respectively. More particularly, Theorems 4 and 7 feature the strong quantifier order $(\exists\mathcal{C})(\forall$ probabilities $p < 1)[\cdots]$; whereas, Theorems 3 and 6 essentially feature the weaker order $(\forall$ probabilities $p < 1)(\exists\mathcal{C})[\cdots]$. There are, however, some apparent costs for the stronger quantifier order of Theorems 4 and 7: the PM's in the proofs of Theorems 4 and 7 are not blind, and, as we shall see, they are not as memory limited as the PM's in the proofs of Theorems 3 and 6. It is open whether these costs are necessary.

[14, (6), Page 367] shows that any indexed class of total computable players is learnable (by a total computable player), and the learning strategy employed is an enumeration strategy [10]. In Section 4 we show (Theorem 18) that *any* total computable player P is (extensionally) equivalent to a player E_P based on enumeration strategy for a suitable indexed class of blind players. The enumeration strategy based E_P can and should be thought of as a *canonical normal form of* P. The canonical computable coordinator E_P corresponding to a pre-given total computable coordinator P may, though, take exponentially more time than P (Theorem 22 also in Section 4), but, of course, the point of the existence of E_P (and the "converse" result [14, (6), Page 367]) is the interesting insight that total coordinators can be *extensionally* conceptualized as exploiting an enumeration technique. From Theorem 22, though, *intensionally* a total coordinator P can be quite different, e.g., in run time, from its (enumeration technique based) canonical form E_P.

TSCOPE(P) denotes the class of total coordinators learnable by a total coordinator P and represents the *competence* of P. In [14] it is shown that the competence, [14, Page 369], of a total player cannot be strictly improved by another total player. In Section 5 we prove that the competencies of any two total players are the same modulo isomorphism. More specifically, Theorem 27 of Section 5 says that, for every pair P and Q of total players, the sets TSCOPE(P) and TSCOPE(Q) are homeomorphic in Cantor space [15, Page 424] (computably homeomorphic, if P and Q are computable). We also investigate TSCOPE(P) from the points of view of of topology and descriptive set theory. We show for instance (Theorem 23) that TSCOPE(P) is an uncountable, dense, meager and measure zero F_σ set of Cantor space, which is Wadge-complete with respect to the class of all F_σ sets of Cantor space.

Finally we provide a completely *effective version* of the competency isomorphism theorem. Let IND(P) denote the index set of the restriction of TSCOPE(P) to computable players for a computable coordinator P. Corollary 30 of Section 5 says that IND(P) and IND(Q) are computably isomorphic for every pair P, Q of computable coordinators.

2 Mathematical preliminaries

We denote by $\{0,1\}^*$ the set of all (finite) binary strings. If $\sigma \in \{0,1\}^*$ then $|\sigma|$ denotes the length of σ; $\sigma(i)$, with $i < |\sigma|$, is the i-th bit of σ; $\sigma \upharpoonright n$ denotes the initial segment of σ of length equal to n; (clearly $\sigma \upharpoonright 0$ is the empty string, denoted by $\emptyset$); if $|\sigma| > 0$ then $\sigma^- = \sigma \upharpoonright (|\sigma| - 1)$. The concatenation of two strings σ and τ is denoted by $\sigma \cdot \tau$.

We let $\mathbb{N}$ denote the set of natural numbers. Suppose $p_m(x_1, \ldots, x_m)$ is a computable bijection from $\mathbb{N}^m$ to $\mathbb{N}$. Then $\pi_1^m, \ldots, \pi_m^m$ denote the corresponding computable inverses of p_m, i.e., $\pi_j^m(p_m(x_1, \ldots, x_m)) = x_j$.

A *player* (synonym: a *coordinator*) is a partial function from $\{0,1\}^*$ into $\{0,1\}$. Given two players F and G, define two sequences $R_{F,G}$ and $R_{G,F}$ by induction as follows. For every n, let

$$R_{F,G}(n) = F(R_{G,F} \upharpoonright n) \qquad R_{G,F}(n) = G(R_{F,G} \upharpoonright n),$$

where $R_{F,G}(n)$ is defined if and only if, for every $i < n$ $R_{G,F}(i)$ is defined, and $F(R_{G,F} \upharpoonright n)$ is defined; similarly for $R_{G,F}(n)$. We say that F and G *coordinate* or that F *learns* G, or else that G *learns* F if and only if both $R_{F,G}$ and $R_{G,F}$ are infinite strings and for almost all i, $R_{F,G}(i) = R_{G,F}(i)$. A *total player* is simply a player which is total. For example, the identically zero coordinator and the identically one coordinator are total, but neither learns the other. Any two total coordinators which output but finitely many ones, do learn to coordinate with each other, but, if one outputs but finitely many ones and the other outputs but finitely many zeros, neither learns the other. Deterministic players will usually be denoted by upper case Latin letters.

A class $\mathcal{C}$ of players is said to be *learnable by a player* F if and only if F learns all elements of $\mathcal{C}$: in this case, we say that F *learns* $\mathcal{C}$ or F *coordinates* $\mathcal{C}$.

We observe that alternatively a player can also be viewed as a function from $\{0,1\}^*$ into $\{0,1\}^*$. Indeed, any player F originates a function from $\{0,1\}^*$ into $\{0,1\}^*$ as follows: if $F(\sigma \upharpoonright j)$ converges for all $j \leq |\sigma|$, then F maps σ into a string $F[\sigma]$ (with $|F[\sigma]| = |\sigma| + 1$) such that for every $i < |\sigma| + 1$, $F[\sigma](i) = F(\sigma \upharpoonright i)$; $F[\sigma]$ is undefined if $F(\sigma \upharpoonright j)$ diverges for some $j \leq |\sigma|$. $F[\sigma]$ denotes the sequence τ (of length $|\sigma| + 1$) that F outputs while coordinating with some player which starts with σ when coordinating with F. One can similarly define $F[\tau]$ for infinite sequences τ.

Conversely, every monotonic partial function $h : \{0,1\}^* \longrightarrow \{0,1\}^*$ such that $|h[\sigma]| = |\sigma| + 1$ (again denoting by $h[\sigma]$ the image of σ under h) can be viewed as a player, by letting $h(\sigma) = h[\sigma](|\sigma|)$.

Throughout the paper we will refer to some effective listing $\{M_i\}_{i \in \mathbb{N}}$ of all (partial) computable deterministic players.

A *probabilistic player* is a player, with the ability to toss coins. Probabilistic players will usually be denoted by PM (with or without subscripts).

For probabilistic coordination, we assume the following model. Consider a probabilistic player PM, which (i) takes as input *two* equal length strings (over $\{0,1\}^*$),[2] (ii) does its computation probabilistically (i.e. it has the ability to toss a fair coin), and (iii) outputs (if defined) either 0 or 1.

[2] In the probabilistic case we need two inputs; whereas, in the deterministic case we needed only one. In the case of M deterministic, M's outputs can be determined from the other machine's data – so M's prior outputs did not need to be part of M's input. In the case of a probabilistic player PM, PM's responses are the second input.

We consider $PM(\sigma, \tau)$, as being the next output of the probabilistic coordinator PM during coordination game, when the past output from the other coordinator has been σ and the past output of PM has been τ. [3] $prob(PM(\sigma, \tau) = b)$ denotes the probability that the output of $PM(\sigma, \tau)$ is b. Now we may define probability over $PM[\sigma]$ naturally as follows. $prob(PM[\emptyset] = a) = prob(PM(\emptyset, \emptyset) = a)$. For $n > 0$, $prob(PM[b_0 \ldots b_n] = a_0 \ldots a_n a_{n+1}) = prob(PM(b_0 \ldots b_n, a_0 \ldots a_n) = a_{n+1}) * prob(PM[b_0 \ldots \ldots b_{n-1}] = a_0 \ldots a_n)$.

Probability of PM coordinating with a deterministic player M may be defined as follows (we will not need to consider coordination among two probabilistic players in this paper). Let

$$f_\sigma(n) = \begin{cases} M[\sigma](n), & \text{if } n < |\sigma|; \\ M[f_\sigma \upharpoonright n](n), & \text{otherwise.} \end{cases}$$

$$g_\sigma(n) = \begin{cases} \sigma(n), & \text{if } n < |\sigma|; \\ M[f_\sigma \upharpoonright n](n), & \text{otherwise.} \end{cases}$$

Intuitively, if M coordinates with another coordinator after seeing σ as input, then f_σ and g_σ respectively denote the sequence output by M and the other coordinator.

Now, probability that PM and M coordinate after PM has output σ is given by, $ProbCd(PM, M, \sigma) = [prob(PM[M[\sigma] \upharpoonright (|\sigma| - 1)] = \sigma)] * \Pi_{n=|\sigma|}^{\infty}[prob(PM(f_\sigma \upharpoonright n, g_\sigma \upharpoonright n) = f_\sigma(n))]$.

Let $CdSet(PM, M) = \{\sigma \mid \sigma = \emptyset \text{ or } \sigma(|\sigma| - 1) \neq M[\sigma](|\sigma| - 1)\}$. Now, the probability of PM coordinating with M can be given by $\sum_{\sigma \in CdSet(PM,M)} ProbCd(PM, M, \sigma)$ (here we take sum of infinitely many 0s as 0).

Let $\{PM_i\}_{i \in \mathbb{N}}$ be some effective listing of all (partial) computable probabilistic players.

3 Probabilistic vs. deterministic coordinators

N.B. Throughout *this* section all of our players are *algorithmic*; also all our random bits are *uniformly distributed*, e.g., from fair coin tosses.

[3] One may more generally allow PM to use its past coin tosses made during the output of τ too. All of our results and proofs, except for Theorem 8 carry over easily to this generalization. Theorem 8 also holds (with slightly different proof) as long as our definition of coordination allows that machine M not coordinate with a probabilistic machine just by being undefined at some point in the coordination process.

The first result in this section is just a lift of a result from [14].

Theorem 1 *For all $n > 1$, there exist classes $\mathcal{C}_1, \ldots, \mathcal{C}_n$ of coordinators such that:*

(a) For each i, $1 \le i \le n$, all members of $\mathcal{C}_i$ coordinate with each other; and
(b) If any coordinator M (even outside above classes) coordinates with all of $\mathcal{C}_i$, then M cannot coordinate with any member of $\bigcup_{1 \le j \le n, j \ne i} \mathcal{C}_j$.

Proof. This can be handled by straightforwardly generalizing the proof for the $n = 2$ case in [14]. ∎

In a sense, then, $\mathcal{C}_1, \ldots, \mathcal{C}_n$ from Theorem 1 are *incompatible* "camps" of computable coordinators.

Definition 2 [14] A player Q is *blind* if $Q(\sigma) = Q(\tau)$ for all $\sigma, \tau \in \{0, 1\}^*$ such that $|\sigma| = |\tau|$.

The remaining results of this section concern the power of probabilistic coordinators, deterministic coordinators, or teams thereof.

We say a coordinator is *k-memory limited* iff it depends functionally only on the (up to) k last bits of its input.

The next four theorems are among the main results of this section mentioned in Section 1 above.

Theorem 3 *Suppose $0 \le p < 1$. There exists a class of computable deterministic players $\mathcal{C}$ such that:*

(a) No computable deterministic coordinator can coordinate with all of $\mathcal{C}$; and
(b) For k chosen large enough that $1 - 2^{-k} \ge p$, there exists a blind *computable probabilistic coordinator PM, such that, for each member M of $\mathcal{C}$, PM can coordinate with M with probability $1 - 2^{-k} \ge p$.*

Interpretation. PM of Theorem 3 just above succeeds in coordinating with the class $\mathcal{C}$ of deterministic players with probability (at least) p, but *no* computable deterministic coordinator can coordinate with every player in $\mathcal{C}$. Hence, probabilistic coordinators beat deterministic ones! Furthermore, PM is blind, i.e., it depends functionally only on the length of the bit strings it sees.

Proof. As above, let k be chosen large enough that $1 - 2^{-k} \ge p$. Let M_n be the n-th computable deterministic player. Define player F_n as follows. Let τ

be any infinite sequence (over 0, 1). $F_n[\tau]$ is defined as follows.

$$F_n[\tau](r) = \begin{cases} 1, & \text{if } r < n; \\ 0, & \text{if } r = n; \\ 1, & \text{if } n < r \leq n+k+1; \\ 1, & \text{if } r \geq n+k+2, \text{ and } (M_n[1^n01^k] \\ & \text{does not halt within } r \text{ steps, or} \\ & \tau \text{ does not start with } M_n[1^n01^k]); \\ 1 - M_n[F_n[\tau \restriction (r-1)]](r), & \text{otherwise.} \end{cases}$$

Note that F_n does not coordinate with M_n. Thus no computable deterministic player can coordinate with $\mathcal{C} = \{F_i \mid i \in \mathbb{N}\}$.

Now define PM as follows. $PM[\tau] = [k \text{ random bits}]1^{|\tau|-k+1}$.

Note that if the random k bits chosen by PM are such that $PM[1^n01^k] \neq M_n[1^n01^k]$, then PM coordinates with F_n. Thus, PM coordinates with each F_n with probability at least $1 - 2^{-k}$. ∎

Furthermore, we can see from the proof just above of Theorem 3 that PM employs only k random bits and is k-memory limited, i.e., it depends functionally only on the (up to) k last bits of its input. This is only so it can keep track if it is to output one of its first k bits, its only random bits.

As mentioned in Section 1, the next theorem provides a variant of Theorem 3 with significantly stronger quantifier order: $(\exists\mathcal{C})(\forall$ probabilities $p < 1)[\cdots]$ instead of $(\forall$ probabilities $p < 1)(\exists\mathcal{C})[\cdots]$. Our witnessing PM is no longer blind.

Theorem 4 *There exists a class of computable deterministic players $\mathcal{C}$ such that:*

(a) No computable deterministic coordinator can coordinate with all of $\mathcal{C}$; and
(b) For all p such that $0 \leq p < 1$, for k chosen large enough that $1-2^{-k} \geq p$, there exists a computable probabilistic coordinator PM such that, for each member M of $\mathcal{C}$, PM can coordinate with M with probability $1-2^{-k} \geq p$.

Proof. Let M_n be the n-th computable deterministic player. Define coordinator F_n as follows. Let τ be any infinite sequence (over 0, 1). $F_n[\tau]$ is defined as follows.

$$F_n[\tau](r) = \begin{cases} 1, & \text{if } r < n; \\ 0, & \text{if } r = n; \\ 1, & \text{if } n < r \leq n+n+1; \\ 1, & \text{if } r \geq n+n+2, \text{ and } (M_n[1^n01^n] \\ & \text{does not halt within } r \text{ steps, or } \tau \\ & \text{does not start with } M_n[1^n01^n]); \\ 1 - M_n[F_n[\tau \restriction (r-1)]](r), & \text{otherwise.} \end{cases}$$

Note that F_n does not coordinate with M_n. Thus no computable deterministic player can coordinate with $\mathcal{C} = \{F_i \mid i \in \mathbb{N}\}$.

Now let $p < 1$ be given. Let k be chosen large enough that $1 - 2^{-k} \geq p$. For $i \leq k$, let $b_i \in \{0, 1\}$ be such that $M_i[1^i0] \neq 1^{i+1}b_i$. Now define PM as follows.

$$PM[\tau] = \begin{cases} 1^{i+1}b_i1^{|\tau|-i-1}, & \text{if } \tau \text{ starts with } 1^i0 \text{ for some } i \leq k; \\ 1^{i+1}[k \text{ random bits}]1^{|\tau|-k-i}, & \text{if } \tau \text{ starts with } 1^i0 \text{ for some } i > k; \\ 1^{|\tau|+1}, & \text{otherwise.} \end{cases}$$

Note that PM coordinates with F_i, $i \leq k$, due to the first clause above. Also due to the second clause, for $i > k$, PM coordinates with F_i with probability at least $1 - 2^{-k}$, as there is at most one set of k-random bits chosen which would make $PM[1^i01^i]$ same as $M_i[1^i01^i]$. ∎

Furthermore, from the proof of Theorem 4, it can be seen that this theorem's PM employs only k random bits and is *not* k-memory limited before the first 0 it sees since it needs to remember how many 1's it saw before the first 0. However, right after the first 0 it sees, it outputs either a deterministic bit or k random bits and no more random bits. Hence, after the first 0 it sees, it becomes k-memory limited from there onward.

Not considered in the last two theorems above is the case of probability $p = 1$. The next result handles this case.

Fact 5 *There exists a class of total computable coordinators, $\mathcal{C}$, such that no deterministic total computable coordinator can coordinate with all of $\mathcal{C}$, but some probabilistic, computable coordinator coordinates with each member of $\mathcal{C}$ with probability 1.*

However, this probabilistic coordinator necessarily *requires* infinitely *many random bits. It is also not blind.*

Proof. Let M_n be the n-th computable deterministic player. Define player F_n

as follows. Let τ be any infinite sequence (over 0, 1). $F_n[\tau]$ is defined as follows.

$$F_n[\tau](r) = \begin{cases} 1, & \text{if } r < n; \\ 0, & \text{if } r = n; \\ 1, & \text{if } M_n[F_n[\tau \upharpoonright (r-2)]] \neq \tau \upharpoonright r; \\ 1 - M_n[F_n[\tau \upharpoonright (r-1)]](r), & \text{otherwise.} \end{cases}$$

Note that, if M_n is total, then F_n is total and F_n does not coordinate with M_n. Thus no total deterministic player can coordinate with $\mathcal{C} = \{F_i \mid M_i$ is total $\}$.

On the other hand the following PM can coordinate with $\mathcal{C}$ with probability one.

Define PM as follows. For any infinite sequence τ over $\{0, 1\}$,

$$PM[\tau](r) = \begin{cases} 1, & \text{if } \tau(r-1) = 1; \\ b, & \text{otherwise, where } b \text{ is a random bit.} \end{cases}$$

Now consider any $F_n \in \mathcal{C}$ (thus M_n is total). Note that PM coordinates with $F_n \in \mathcal{C}$ as long as F_n, while coordinating with PM, outputs only finitely many 0s. By definition of F_n, this would happen if PM's response to F_n differs from M_n's response to F_n at least at one place. Thus it suffices to show that, when τ contains infinitely many 0s, PM's response to τ differs from M_n's response to τ with probability one. This holds as M_n is deterministic and anytime PM sees a 0 in the last bit of the input, it outputs a random bit.

It follows that PM coordinates with F_n with probability one. ∎

Note that need of infinitely many random bits in above fact is necessary. (If PM can coordinate with a class $\mathcal{C}$ with probability one, using only finitely many random bits, then it must coordinate with $\mathcal{C}$ on all possible random bits, and in particular when the random bits are all 0. Thus, a deterministic coordinator can coordinate with all of $\mathcal{C}$).

It is open whether the *non-blindness* of the probabilistic coordinator in Fact 5 just above is necessary.

For $1 \leq i \leq m$, $\mathbf{Team}^i_m\mathbf{Coord}$ denotes the collection of classes $\mathcal{C}$ of algorithmic *deterministic* players such that we have m deterministic algorithmic coordinators $M^1, \ldots, M^m$ so that for each element F of $\mathcal{C}$, at least i of $M^1, \ldots, M^m$ coordinate with F.

Theorem 6 *Suppose* $0 \leq p < 1$, *and* $m \in \mathbb{N}^+$. *Let* k *be chosen large enough that* $\frac{2^k - m}{2^k} \geq p$.

Then there exists a class of computable deterministic players $\mathcal{C}$ such that:

(a) $\mathcal{C} \notin \mathbf{Team}^1_m\mathbf{Coord}$;
(b) There are $m+1$ blind deterministic machines such that, these machines witness that $\mathcal{C} \in \mathbf{Team}^1_{m+1}\mathbf{Coord}$.
(c) There exists a computable probabilistic coordinator PM, such that, for each member M of $\mathcal{C}$, PM can coordinate with M with probability $\frac{2^k-m}{2^k} \geq p$.

Interpretation. PM of Theorem 6 just above succeeds in coordinating with the class $\mathcal{C}$ of deterministic players with probability (at least) p. However, $\mathcal{C} \in (\mathbf{Team}^1_{m+1}\mathbf{Coord} - \mathbf{Team}^1_m\mathbf{Coord})$. Therefore, probabilistic coordination is *not* characterized by deterministic team coordination. $\mathcal{C} \in \mathbf{Team}^1_{m+1}\mathbf{Coord}$ is witnessed by $m+1$ *blind* deterministic coordinators.

Proof. As above, let k be chosen large enough that $\frac{2^k-m}{2^k} \geq p$.

Let M_i be the i-th computable deterministic player. Define coordinator F_n as follows. Let τ be any infinite sequence (over 0, 1). $F_n[\tau]$ is defined as follows.

$$F_n[\tau](r) = \begin{cases} 1, & \text{if } r < n; \\ 0, & \text{if } r = n; \\ 1, & \text{if } n < r \leq n+k+1; \\ 1, & \text{if } r \geq n+k+2, \text{ and } \\ & (\forall j \mid 1 \leq j \leq m) \\ & [M_{\pi^m_j(n)}[1^n01^k] \text{ does not halt} \\ & \text{within } r \text{ steps or } \tau \text{ does not} \\ & \text{start with } M_{\pi^m_j(n)}[1^n01^k]]; \\ 1 - M_{\pi^m_j(n)}[F_n[\tau \upharpoonright (r-1)]](r), & \text{not above cases, and} \\ & M_{\pi^m_j(n)}[F_n[\tau \upharpoonright (r-1)]]\downarrow \supseteq \tau \upharpoonright r \\ & \text{for first such } j,\ 1 \leq j \leq m, \\ & \text{found.} \end{cases}$$

(a) Note that F_n does not coordinate with $M_{\pi^m_j(n)}$, for $1 \leq j \leq m$. Thus $\mathcal{C} = \{F_i \mid i \in \mathbb{N}\} \notin \mathbf{Team}^1_m\mathbf{Coord}$.

(b) To show that $\mathcal{C} \in \mathbf{Team}^1_{m+1}\mathbf{Coord}$, consider machines $M^1, \ldots, M^{m+1}$ such that $M^i[\tau] = w_i 1^{|\tau|-|w_i|+1}$, where $w_1, w_2, \ldots, w_{m+1}$ are m distinct strings over $\{0,1\}$ of length $\lceil \log_2(m+1) \rceil$ each. We claim that for each F_n, at least one of these machines coordinates with F_n. This is so since at most m of $M^i(F_n)$ can start with one of $M_{\pi^m_j(n)}[1^n01^n]$, $1 \leq j \leq n$. Thus, at least one of the machines M^i coordinates with F_n.

(c) Define probabilistic machine PM as follows.

$PM[\tau] = [k \text{ random bits }]1^{|\tau|-k+1}$. Note that if the random k bits chosen by PM are such that $PM[1^n01^k] \neq M_{\pi_j^m(n)}[1^n01^k]$, for $1 \leq j \leq m$, then PM coordinates with F_n. Thus, PM coordinates with each F_n with probability at least $\frac{2^k-m}{2^k}$. $\blacksquare$

Furthermore, from the proof of Theorem 6 just above, it can be seen that this theorem's PM employs only k random bits and *is* k-memory limited. Also, each of the $m+1$ blind deterministic coordinators witnessing $\mathcal{C} \in \mathbf{Team}^1_{m+1}\mathbf{Coord}$ is $\lceil \log_2(m+1) \rceil$-memory limited.

As mentioned in Section 1, the next theorem provides a variant of Theorem 6 with significantly stronger quantifier order. Our witnessing PM is no longer blind.

Theorem 7 *Suppose $m \in \mathbb{N}^+$. Then there exists a class of computable deterministic players $\mathcal{C}$ such that:*

(a) $\mathcal{C} \notin \mathbf{Team}^1_m\mathbf{Coord}$*;*
(b) There are $m+1$ blind computable deterministic machines witnessing $\mathcal{C} \in \mathbf{Team}^1_{m+1}\mathbf{Coord}$; and
(c) For all p such that $0 \leq p < 1$, for k chosen large enough that $\frac{2^k-m}{2^k} \geq p$, there exists a computable probabilistic coordinator PM such that, for each member M of $\mathcal{C}$, PM can coordinate with M with probability $\frac{2^k-m}{2^k} \geq p$.

Proof. We use a similar modification of Proof of Theorem 6, as was done for Proof of Theorem 3 to get Theorem 4.

Details are as follows:

Let M_i be the i-th computable deterministic player. Define coordinator F_n as follows. Let τ be any infinite sequence (over 0, 1). $F_n[\tau]$ is defined as follows.

$$F_n[\tau](r) = \begin{cases} 1, & \text{if } r < n; \\ 0, & \text{if } r = n; \\ 1, & \text{if } n < r \leq n+n+m+1; \\ 1, & \text{if } r \geq n+n+m+2, \text{ and} \\ & (\forall j \mid 1 \leq j \leq m) \\ & [M_{\pi_j^m(n)}(1^n01^{n+m}) \text{ does not halt} \\ & \text{within } r \text{ steps or } \tau \text{ does not} \\ & \text{start with } M_{\pi_j^m(n)}(1^n01^{n+m})]; \\ 1 - M_{\pi_j^m(n)}[F_n[\tau \restriction (r-1)]](r), & \text{not above cases and} \\ & M_{\pi_j^m(n)}[F_n[\tau \restriction (r-1)]]\downarrow \supseteq \tau \restriction r; \\ & \text{for first such } j,\ 1 \leq j \leq m, \\ & \text{found.} \end{cases}$$

(a) Note that F_n does not coordinate with $M_{\pi_j^m(n)}$, for $1 \leq j \leq m$. Thus $\mathcal{C} = \{F_i \mid i \in \mathbb{N}\} \notin \mathbf{Team}_m^1\mathbf{Coord}$.

(b) To show that $\mathcal{C} \in \mathbf{Team}_{m+1}^1\mathbf{Coord}$, consider machines $M^1, \ldots, M^{m+1}$ such that $M^i[\tau] = w_i 1^{|\tau|-|w_i|+1}$, where $w_1, w_2, \ldots, w_{m+1}$ are m distinct strings over $\{0,1\}$ of length $\lceil \log_2(m+1) \rceil$ each. We claim that for each F_n, at least one of these machines coordinates with F_n. This is so since at most m of $M^i(F_n)$ can start with one of $M_{\pi_j^m(n)}[1^n01^{n+m}]$, $1 \leq j \leq n$. Thus, at least one of the machines M^i coordinates with F_n.

(c) Suppose $0 \leq p < 1$ is given. Let k be chosen large enough that $\frac{2^k - m}{2^k} \geq p$. For $i \leq k$, let $b_i \in \{0,1\}^m$ be such that $M_{\pi_j^m(i)}[1^i01^{i+m}]$ does not start with $1^{i+1}b_i$, for $1 \leq j \leq m$. Now define PM as follows.

$$PM[\tau] = \begin{cases} 1^{i+1}b_i1^{|\tau|-i-m}, & \text{if } \tau \text{ starts with } 1^i0 \text{ for some } i \leq k; \\ 1^{i+1}[k \text{ random bits}]1^{|\tau|-k-i}, & \text{if } \tau \text{ starts with } 1^i0 \text{ for some } i > k; \\ 1^{|\tau|+1}, & \text{otherwise.} \end{cases}$$

Note that PM coordinates with F_i, $i \leq k$, due to the first clause above. Also due to the second clause above, for $i > k$, M coordinates with F_i with probability at least $\frac{2^k-m}{2^k}$, as there are at most m sets of k-random bits chosen which would make $PM[1^i01^{i+m}]$ same as one of $M_{\pi_j^m(i)}[1^i01^{i+m}]$. ∎

Furthermore, from the proof of Theorem 7 just above, it can be seen that this theorem's PM employs only k random bits and is *not* k-memory limited. However, after the first 0 it sees, it becomes k-memory limited from there onward. Also, each of the $m+1$ blind deterministic coordinators witnessing $\mathcal{C} \in \mathbf{Team}_{m+1}^1\mathbf{Coord}$ is $\lceil \log_2(m+1) \rceil$-memory limited.

For probabilities p, $\mathbf{Prob}_p\mathbf{Coord}$ denotes the collection of classes $\mathcal{C}$ of *deterministic* players such that some probabilistic coordinator coordinates with each element of $\mathcal{C}$ with probability $\geq p$.

Theorem 8 *Suppose $\ell, m \in \mathbb{N}^+$ and $0 \leq p \leq 1$ are given such that $\frac{\ell}{m} < p$. Then, there is a class of computable deterministic players $\in (\mathbf{Team}_m^\ell\mathbf{Coord} - \mathbf{Prob}_p\mathbf{Coord})$.*

Moreover, for k such that $2^k \geq m$, the positive half of this theorem is witnessed by a team of blind, computable deterministic coordinators which are k-memory limited.

Proof. As above, choose k such that $2^k \geq m$. Let M_i be the i-th computable deterministic player. Let PM_i denote the i-th probabilistic player. Define coordinator F_n as follows. Let τ be any infinite sequence (over 0, 1). $F_n[\tau]$ is defined as follows.

Let X be the first m elements of $\{0,1\}^k$. Let S_n^w be the lexicographically least subset Y of X of cardinality ℓ, such that $\text{Prob}(\{PM_n[1^n 0 1^k]$ is defined within w steps, and starts with some $\sigma \in Y\}) \leq \frac{\ell}{m}$. Then:

- if $r \leq n+k$ then let

$$F_n[\tau](r) = \begin{cases} 1, & \text{if } r < n; \\ 0, & \text{if } r = n; \\ 1, & \text{if } n < r \leq n+k. \end{cases}$$

- if $r > n+k$ then search for the first $w \geq r-1$ such that either

(i) τ starts with an element of S_n^w; or

(ii) otherwise and within $w-r-1$ steps, for some $x \in \{0,1\}$, one can verify that $PM_n(F_n[\tau \upharpoonright (r-1)], \tau \upharpoonright r) = x$ with probability at least 1/3.

If such a w is found then let

$$F_n[\tau](r) = \begin{cases} 1, & \text{if (i) holds} \\ 1-x & \text{if (ii) holds for some first such } x \text{ found.} \end{cases}$$

Let $F_n[\tau](r)$ be undefined otherwise.

Let $S_n = \lim_{w\to\infty} S_n^w$. Note that S_n is well defined, as the limit of S_n^w exists.

Let $\mathcal{C} = \{F_n \mid n \in \mathbb{N}\}$.

Now,

(a) The probability that PM_n while interacting with F_n starts with an element of S_n is at most $\frac{\ell}{m}$ (by definition of S_n).

(b) Let $r > n+1+k$ be large enough so that $S_n^w = S_n$, for any $w \geq r$. Now consider any $\sigma \in \{0,1\}^r$ such that σ does not start with any element of S_n. Now, $PM_n(F_n[\sigma \upharpoonright (|\sigma|-1)], \sigma) \neq F_n(\sigma)$ with probability at least 1/3 (due to the diagonalization in construction of F_n (see (ii) above)). It follows that, if PM_n while interacting with F_n does not start with any element of S_n, then the probability of coordination is zero. Thus, the probability that P_n can coordinate with F_n is at most $\frac{\ell}{m}$. It follows that $\mathcal{C} \notin \mathbf{Prob}_p\mathbf{Coord}$.

(c) F_n coordinates with any infinite sequence τ which starts with element of S_n and converges to 1. Define m machines $M^1, \ldots, M^m$, such that $M^i[\tau] = \tau_i 1^{|\tau|-|\tau_i|+1}$, for any τ, where τ_i is the i-th string in X. It follows that, for any n, at least $\text{card}(S_n) = l$ of the machines $M^1, \ldots, M^m$, coordinate with F_n.

Theorem follows. ∎

The next corollary is one of the main results of this section mentioned in

Section 1 above. It shows that the power of probabilistic coordination of deterministic classes of players is strict in the associated probability.

Corollary 9 *Suppose* $0 \leq p < q \leq 1$. *Then, there is a class of deterministic players* $\in$ ($\mathbf{Prob}_p\mathbf{Coord} - \mathbf{Prob}_q\mathbf{Coord}$). *Moreover, for* ℓ, m *such that* $p \leq \frac{\ell}{m} < q$ *and* $k = \lceil \log_2(m) \rceil$, *the positive half of this corollary is witnessed by a blind probabilistic coordinator which employs only* k *random bits and is* k*-memory limited.*

The next two results together completely characterize the relative power of the coordination classes of the form $\mathbf{Team}^i_m\mathbf{Coord}$.

Theorem 10 *Suppose* $\ell, m, v, w \in \mathbb{N}^+$ *such that* $\ell \leq m$, $v \leq w$, *and there is no way to distribute* w *balls among* m *boxes such that any combination of* ℓ *boxes receives at least* v *balls. Then,* $\mathbf{Team}^\ell_m\mathbf{Coord} - \mathbf{Team}^v_w\mathbf{Coord} \neq \emptyset$.

Proof. Let k be such that $2^k \geq m$. Let M_i be the i-th deterministic player. Define player F_n as follows. Let τ be any sequence (over 0, 1). $F_n[\tau]$ is defined as follows.

Let X be the first m elements of $\{0,1\}^k$. Let S^t_n be the lexicographically least subset Y of X of cardinality ℓ, such that card($\{j \mid 1 \leq j \leq w$ and $M_{\pi^w_j(n)}[1^n01^k]$ halts within t steps, and starts with $\sigma \in Y\}) < v$.

Define $F_n[\tau]$ as follows:

- if $r \leq n + k$ then let

$$F_n[\tau](r) = \begin{cases} 1, & \text{if } r < n; \\ 0, & \text{if } r = n; \\ 1, & \text{if } n < r \leq n + k. \end{cases}$$

- if $r > n + k$ then search for the first $t \geq r - 1$ such that either

(i) τ starts with an element of S^t_n; or

(ii) otherwise and within $t - r - 1$ steps, one can find an x such that $M_{\pi^w_j(n)}[F_n[\tau \upharpoonright (r-1)]]\downarrow \supseteq (\tau \upharpoonright r) \cdot x$ for some j.

  If such a t is found then let

$$F_n[\tau](r) = \begin{cases} 1, & \text{if (i) holds;} \\ 1 - x & \text{if (ii) holds for some first such } x, j. \end{cases}$$

  Let $F_n[\tau](r)$ be undefined otherwise.

Let $S_n = \lim_{t \to \infty} S^t_n$. Note that S_n is well defined, as the limit of S^t_n exists.

Let $\mathcal{C} = \{F_n \mid n \in \mathbb{N}\}$.

Now,

(a) Let $r > n + 1 + k$ be large enough so that $S_n^t = S_n$, for any $t \geq r$. Now consider any $\sigma \in \{0,1\}^r$ such that σ does not start with any element of S_n. Then by the diagonalization clause in the construction of F_n, we have that none of the machines in $M_{\pi_1^w(n)}, \ldots, M_{\pi_w^w(n)}$, which starts with σ while interacting with F_n, can coordinate with F_n.

On the other hand, the number of machines in $M_{\pi_1^w(n)}, \ldots, M_{\pi_w^w(n)}$, which start with $\sigma \in S_n$ while interacting with F_n is $< v$. Thus, $\{M_{\pi_j^w(n)} \mid 1 \leq j \leq w\}$, do not witness that $\mathcal{C} \in \mathbf{Team}_w^v\mathbf{Coord}$.

(b) F_n coordinates with any infinite sequence τ which starts with an element of S_n and converges to 1. Define m machines $M^1, \ldots, M^m$, such that $M^i[\tau] = \tau_i 1^{|\tau|-|\tau_i|+1}$, for any τ, where τ_i is the i-th string in X. It follows that, for any n, at least $\text{card}(S_n) = \ell$ of the machines $M^1, \ldots, M^m$, coordinate with F_n.

Theorem follows. ∎

Proposition 11 *Suppose $\ell, m, v, w \in \mathbb{N}^+$ such that $\ell \leq m$, $v \leq w$, and there exists a way to distribute w balls among m boxes such that any combination of l boxes receives at least v balls. Then,* $\mathbf{Team}_m^\ell\mathbf{Coord} \subseteq \mathbf{Team}_w^v\mathbf{Coord}$.

Proof. Assume the boxes and balls in above to be numbered (from 1 to m and 1 to w respectively). Now suppose $M^1, \ldots, M^m$ are given. Define $M'^{,1}, \ldots, M'^{,w}$ such that $M'^{,i}$ follows M^j, if the i-ball in the above distribution falls in j-th box. Now it is easy to verify that the players which can be $\mathbf{Team}_m^\ell\mathbf{Coord}$ identified by $M^1, \ldots, M^m$ are also $\mathbf{Team}_w^v\mathbf{Coord}$ identified by $M'^{,1}, \ldots, M'^{,w}$. ∎

Corollary 12 *Suppose $\ell, m, v, w \in \mathbb{N}^+$ such that $\ell \leq m$, $v \leq w$. Then,* $\mathbf{Team}_m^\ell\mathbf{Coord} \subseteq \mathbf{Team}_w^v\mathbf{Coord}$ *iff there exists a way to distribute w balls among m boxes such that any combination of l boxes receives at least v balls.*

4 Normal form characterization of learnable classes of computable players

N.B. Throughout *this* section players are always *total computable.*

Definition 13 An *indexed class* of computable binary functions (*indexed class* for short) is a class $\mathcal{C}$ of computable binary functions such that there is a total computable function $C(i,x)$ from $\mathbb{N}^2$ into $\{0,1\}$ such that

$$\mathcal{C} = \{\lambda x.\, C(i,x) \mid i \in \mathbb{N}\}.$$

In this case, $C(i,x)$ is said to be an *enumerating function* for $\mathcal{C}$.

Definition 14 [1,2] We say that a total function P, NV-identifies a function f, iff for all but finitely many n, $P(f \upharpoonright n) = f(n)$.

Definition 15 Let $\mathcal{C}$ be an indexed and dense class ("dense" means that for every $\sigma \in \{0,1\}^*$ there is some $f \in \mathcal{C}$ such that $\sigma \subset f$), and let $C(i,x)$ be an enumerating function for $\mathcal{C}$. We define a total computable function $\mathrm{LE}(C)$ from $\{0,1\}^*$ into $\{0,1\}$ as follows. Given $\sigma \in \{0,1\}^*$, let

$$\mathrm{LE}(C)(\sigma) = C(i(\sigma), |\sigma|)$$

where

$$i(\sigma) = \min(\{i \in \mathbb{N} : (\forall j < |\sigma|)[\sigma(j) = C(i,j)]\})$$

(such an i exists because $\mathcal{C}$ is dense).

Note that $\mathrm{LE}(C)$ is an algorithm which NV-identifies $\mathcal{C}$ by enumeration: given σ, $\mathrm{LE}(C)$ first finds the first i such that $\lambda x.C(i,x)$ is consistent with σ, and then outputs the next value $C(i, |\sigma|)$ of $\lambda x.C(i,x)$. Of course, $\mathrm{LE}(C)$ is also a total computable player. Note also that $\mathrm{LE}(C)$ depends on the enumeration function $C(i,x)$ and not only on $\mathcal{C}$.

Definition 16 Let $\sigma \in \{0,1\}^*$, and let P be a total computable player. We define a total computable function P_σ from $\mathbb{N}$ into $\{0,1\}$ as follows:

$$P_\sigma(n) = \begin{cases} \sigma(n) & \text{if } n < |\sigma|, \\ P(P_\sigma \upharpoonright n) & \text{otherwise.} \end{cases}$$

Definition 17 Let $\mathcal{C}$ and $C(i,x)$ be as in Definition 15. Define

$$\mathcal{P}(C) = \{P \mid P \text{ is a total player and } (\exists \sigma \in \{0,1\}^*)[R_{P,\mathrm{LE}(C)} = \mathrm{LE}(C)_\sigma]\}$$

(where we identify $\mathrm{LE}(C)_\sigma$ with the infinite string of its values).

Roughly speaking, $\mathcal{P}(C)$ consists of all players which, as far as the opponent behaves like $\mathrm{LE}(C)$, behave as $\mathrm{LE}(C)_\sigma$ for a suitable $\sigma \in \{0,1\}^*$. A moment's reflection shows that $\mathcal{P}(C)$ is precisely the class of all total computable players which are learned by $\mathrm{LE}(C)$.

Theorem 18 next is a main result of this section mentioned above in Section 1. The player $\mathrm{LE}(C_P)$ of Theorem 18 can and should be thought of as an enumeration strategy based *canonical normal form of P*. This also yields another corroborating evidence of the thesis [12] that for each type of Gold-style learning, there is an adequate enumeration technique.

Theorem 18 *(a) For every total computable player P there are a dense indexed class $\mathcal{C}_P$ and an enumerating function $C_P(i,x)$ for $\mathcal{C}_P$ such that $P = \mathrm{LE}(C_P)$.*

(b) *Hence a class $\mathcal{D}$ of total computable players is learnable by a total computable player if and only if there are a dense indexed class $\mathcal{C}$ and an enumerating function $C(i,x)$ for $\mathcal{C}$ such that $\mathcal{D} \subseteq \mathcal{P}(C)$.*

Proof. (a) Let P be a total computable player. Let $\{\sigma_i\}_{i\in\mathbb{N}}$ be a 1-1 computable numbering of all binary strings, such that $|\sigma_i| < |\sigma_j|$ implies $i < j$. Given σ let $\sigma^\sharp$ denote the unique i such that $\sigma = \sigma_i$. Let $C_P(i,n) = P_{\sigma_i}(n)$. Clearly, $C_P(i,n)$ is a total computable function from $\mathbb{N}^2$ into $\{0,1\}$. Let $f_i = \lambda n.C_P(i,n)$, and let $\mathcal{C}_P = \{f_i : i \in \mathbb{N}\}$. We prove (a) by means of the following claims.

(i) $\mathcal{C}_P$ is a dense indexed class. Indeed, $\mathcal{C}_P$ is clearly an indexed class, and for all $\sigma \in \{0,1\}^*$, $f_{\sigma^\sharp}$ extends σ.

(ii) If $\sigma \subseteq \tau \subseteq P_\sigma$, then $P_\tau = P_\sigma$. Indeed, if $\sigma \subseteq \tau \subseteq P_\sigma$ then we can prove by induction on n, that for every n, $P_\tau(n) = P_\sigma(n)$. Suppose the claim holds for all $i < n$. If $n < |\tau|$, then $P_\tau(n) = \tau(n) = P_\sigma(n)$, as $\tau \subseteq P_\sigma$. Otherwise,

$$P_\tau(n) = P(P_\tau \restriction n) \overset{\mathrm{IH}}{=} P(P_\sigma \restriction n) = P_\sigma(n).$$

(The equality $\overset{\mathrm{IH}}{=}$ is justified by the Induction Hypothesis.)

(iii) $\sigma_{i(\sigma)} \subseteq \sigma$, where $i(\sigma) = \min(\{i \in \mathbb{N} : (\forall j < |\sigma|)[C_P(i,j) = \sigma(j)]\})$. Indeed, from the definition of $C_P(i,n)$ it follows that $\sigma_{i(\sigma)}$ is a sequence τ such that $\sigma \subseteq P_\tau$, and $\tau^\sharp \leq \rho^\sharp$ for every $\rho \in \{0,1\}^*$ with $\sigma \subseteq P_\rho$, thus, since $\sigma \subseteq P_\sigma$, one must have $\tau^\sharp \leq \sigma^\sharp$, and $\tau \subseteq \sigma$.

(iv) $P_\sigma = P_{\sigma_{i(\sigma)}}$. This follows by (ii) and (iii) since $\sigma_{i(\sigma)} \subseteq \sigma \subseteq P_{\sigma_{i(\sigma)}}$.

(v) $\mathrm{LE}(C_P) = P$. Indeed, for every $\sigma \in \{0,1\}^*$, one has:

$$\mathrm{LE}(C_P)(\sigma) = C_P(i(\sigma), |\sigma|) = P_{\sigma_{i(\sigma)}}(|\sigma|) = P_\sigma(|\sigma|),$$

(the last equality follows from (iv)), and finally:

$$\mathrm{LE}(C_P)(\sigma) = P_\sigma(|\sigma|) = P(P_\sigma \restriction (|\sigma| - 1)) = P(\sigma).$$

These claims imply part (a).

As regards part (b), let $\mathcal{D}$ be a learnable class of total computable players, and let P be a total computable player that learns $\mathcal{D}$. By claim (a), there are an indexed class $\mathcal{C}_P$ and an enumerating function $C_P(i,x)$ for $\mathcal{C}_P$ such that $P = \mathrm{LE}(C_P)$. Moreover, we observed before that $\mathcal{P}(C_P)$ is the class of all total computable players learned by $\mathrm{LE}(C_P) = P$. Thus $\mathcal{D} \subseteq \mathcal{P}(C_P)$, as desired. ∎

Definition 19 Given a player P, we define:

$\mathrm{BLINDSCOPE}(P) = \{Q : Q$ total computable blind player and P learns $Q\}$.

Definition 20 An *indexed class of players* is a class $\mathcal{C}$ such that there is a total computable function $C(i,\sigma)$ from $\mathbb{N} \times \{0,1\}^*$ into $\{0,1\}$ (called an *enumerating function for $\mathcal{C}$*) such that for all $P \in \mathcal{C}$ there is an $i \in \mathbb{N}$ such that for all $\sigma \in \{0,1\}^*$, $P(\sigma) = C(i, \sigma)$.

Theorem 21 *A class of total computable blind players is learnable by a total computable player if and only if it is contained in an indexed class of players.*

Proof. To any total computable function f from $\mathbb{N}$ into $\{0,1\}$ we associate a blind player Q_f defined by $Q_f(\sigma) = f(|\sigma|)$. We first show that for every total computable player P, BLINDSCOPE(P) = $B(P)$ where

$$B(P) = \{Q_{P_\sigma} : \sigma \in \{0,1\}^*\}.$$

Indeed, if Q is blind then there exists a computable function f such that $Q = Q_f$, and P learns Q if and only if P NV-identifies f. On the other hand, P NV-identifies f if and only if $f = P_\sigma$ for some $\sigma \in \{0,1\}^*$. Thus if Q is blind we have that P learns Q if and only if $Q = Q_{P_\sigma}$ for some σ, i.e. BLINDSCOPE(P) = $B(P)$.

Hence a class of total computable blind players is learnable by a total computable player if and only if it is contained in $B(P)$ for some total computable player P, and $B(P)$ is an indexed class of players, being enumerated by $C_P(i, \tau) = P_{\sigma_i}(|\tau|)$. ∎

The last result in this section provides a complexity upper bound for canonical normal forms for total coordinators — in terms of the complexity of the coordinators being put into normal form.

Let P be a given total coordinator. Let C_P denote the enumerating function of the indexed family $\mathcal{C}_P$ generated for this P, and let $E_P = \mathrm{LE}(C_P)$ be the canonical coordinator provided by Theorem 18 exploiting learning by enumeration for C_P. Clearly,

$$Time(C_P(i,n)) \leq O(\begin{cases} |\sigma_i|, & \text{if } n < |\sigma_i|; \\ \Sigma_{|\sigma_i| \leq j < n} Time(P(\lambda x.C_P(i,x) \upharpoonright j)) & \text{if } n \geq |\sigma_i|. \end{cases})$$

Furthermore, note that in above time, one can calculate not only $C_P(i,n)$, but all of $C_P(i,0), C_P(i,1), \ldots, C_P(i,n)$. Note also that for all σ, there is $i \leq 2^{|\sigma|+1}$, such that $\sigma_i = \sigma$ and $\sigma_{i(\sigma)} \subseteq \sigma$ by (iii) in the proof of Theorem 18 (a). Thus, $E_P(\sigma)$ can be computed in time

$$O(2^{|\sigma|+1} * \max(\{Time(C_P(i, |\sigma|)) \mid i \leq 2^{|\sigma|+1}\})) \tag{1}$$

We get, then, our last main result of this section (mentioned above in Section 1) showing that, while the canonical forms featured above in this section are insightful, running them in place of the originals can be inefficient.

Theorem 22 *Assume the time $Time_P(\cdot)$ used by P is monotonically increasing on length of input and yields the complexity lengthwise (i.e., as the maximum over a particular length of inputs).*

Then, the time to compute $E_P(\sigma)$ is $\leq$

$$O(2^{|\sigma|} * |\sigma| * Time_P(|\sigma|)).$$

Proof. Clearly (1) above is upperbounded by $O(2^{|\sigma|} * |\sigma| * Time_P(|\sigma|))$. ∎

5 Topological and computability theoretic aspects of learnable classes of total players

In this section we investigate learnable classes from the points of view of descriptive set theory and computability theory. All unexplained notions and background material used in this section can be found in [15, Chapter 3]. A total player can be regarded as an element of the topological space $\{0,1\}^{\{0,1\}^*}$ (with respect to the product topology, where $\{0,1\}$ is equipped with the discrete topology), called *Cantor Space*. We recall that in this topological space the clopen (i.e. closed and open) sets form a base: For every finite set $E \subseteq \{0,1\}^*$ and every function $f \in \{0,1\}^E$ let

$$C_f = \{F \in \{0,1\}^{\{0,1\}^*} : (\forall \sigma \in E)[F(\sigma) = f(\sigma)]\};$$

then a set C is a *basic* clopen set if and only if $C = C_f$, for some finite function f as just described. By compactness of the space it turns out that the clopen sets are exactly the finite unions of basic clopens sets.

Given a coordinator P, a sequence $\{A_{m,n}\}_{m,n \in \mathbb{N}}$ of clopen sets is said to be *P-computable*, if there is some index z such that, for any $F \in \{0,1\}^{\{0,1\}^*}$,

$$\varphi_z^{F,P}(m,n) = \begin{cases} 1 & \text{if } F \in A_{m,n}, \\ 0 & \text{otherwise} \end{cases}$$

Here, $\varphi_z^{F,P}$ is the function computed by the oracle Turing machine with index z supplied with an oracle for F and P. If P is computable then we just say that the sequence $\{A_{m,n}\}_{m,n \in \mathbb{N}}$ is *computable*.

In a similar way one defines the notion of a P-computable sequence $\{A_n\}_{n \in \mathbb{N}}$ of clopens, and the notion of a P-computable clopen. A subset $X \subseteq \{0,1\}^{\{0,1\}^*}$ is said to be $\Sigma_2^{0,P}$ if

$$X = \bigcup_m \bigcap_n A_{m,n}$$

where $\{A_{m,n}\}$ is a sequence of clopen sets which is P-computable. If $X \in \Sigma_2^{0,P}$ and P is computable then we just say that X is Σ_2^0. A subset $X \subseteq \{0,1\}^{\{0,1\}^*}$ is said to be F_σ if X is a countable union of closed sets: It turns out that $X \in F_\sigma$ if and only if $X \in \Sigma_2^{0,P}$ for some P. A (total) mapping $\Gamma : \{0,1\}^{\{0,1\}^*} \longrightarrow \{0,1\}^{\{0,1\}^*}$ is called P-*computable* (or simply *computable* if P is computable) if, for any $F \in \{0,1\}^{\{0,1\}^*}$, $\Gamma(F) = \lambda\sigma.\, \varphi_z^{F,P}(\sigma)$, for some index z. We say that Γ is P, Q-*computable* if Γ is $P \oplus Q$-computable, where $P \oplus Q$ is any coordinator obtained by joining together P and Q in some computable way, e.g. $P \oplus Q(\sigma_{2i}) = P(\sigma_i)$ and $P \oplus Q(\sigma_{2i+1}) = Q(\sigma_i)$, where $\{\sigma_i\}_{i \in \mathbb{N}}$ is a 1-1 computable numbering of all binary strings, as in the proof of (a) of Theorem 18. It turns out that a mapping $\Gamma\{0,1\}^{\{0,1\}^*} \longrightarrow \{0,1\}^{\{0,1\}^*}$ is continuous if and only if it is P-computable for some P.

We are interested in the set of total (but not necessarily computable) players learned by a total player P. For every total player P, we define:

$$\mathrm{TSCOPE}(P) = \{Q \in \{0,1\}^{\{0,1\}^*} : P \text{ learns } Q\}.$$

Theorem 23*(i)* TSCOPE(P) *is a* $\Sigma_2^{0,P}$ *subset of* $\{0,1\}^{\{0,1\}^*}$, *and for every oracle* Q *and set* $X \in \Sigma_2^{0,Q}$ *there is a* P, Q*-computable mapping* Γ *such that*

$$X = \Gamma^{-1}(\mathrm{TSCOPE}(P)).$$

It follows that for every F_σ *subset* $X \subseteq \{0,1\}^{\{0,1\}^*}$ *there is a continuous mapping* Γ *such that* $X = \Gamma^{-1}(\mathrm{TSCOPE}(P))$.

(ii) TSCOPE(P) *is dense.*

(iii) TSCOPE(P) *is meager.*

(iv) TSCOPE(P) *has Lebesgue measure zero.*

(v) TSCOPE(P) *has the cardinality of the continuum. In fact,* TSCOPE(P) *intersects every Turing degree* $\mathbf{a}$, *with* $\deg_T(P) \leq_T \mathbf{a}$.

Proof. Let P be given. We prove the claims one by one:

(i). For every $n \in \mathbb{N}$, let

$$A(P,n) = \{Q \in \{0,1\}^{\{0,1\}^*} : R_{Q,P}(n) = R_{P,Q}(n)\}.$$

Then $A(P,n)$ is a P-computable clopen set, and

$$\mathrm{TSCOPE}(P) = \bigcup_n \bigcap_{m \geq n} A(P,m).$$

This easily proves that $\mathrm{TSCOPE}(P) \in \Sigma_2^{0,P}$.

Let now $X \in \Sigma_2^{0,Q}$, say $X = \bigcup_m \bigcap_n A_{m,n}$, where $\{A_{m,n}\}$ is a Q-computable sequence of clopen sets and Q is any oracle. For every $F \in \{0,1\}^{\{0,1\}^*}$ define a function $P_F \in \{0,1\}^{\{0,1\}^*}$ and a sequence $\{m_k\}_{k \in \mathbb{N}}$ as follows. For every $\sigma \in \{0,1\}^*$,

- if $\sigma = \emptyset$ then let $P_F(\sigma) = 0$, and let $m_0 = 0$;
- if $|\sigma| = k + 1$ and we have defined m_k and $P_F(\tau)$ for every τ such that $|\tau| \leq k$, then define

$$P_F(\sigma) = \begin{cases} P(P_F[\sigma^-]) & \text{if } (\forall n \leq k)[F \in A_{m_k,n}], \\ 1 - P(P_F[\sigma^-]) & \text{otherwise.} \end{cases}$$

  Moreover let

$$m_{k+1} = \begin{cases} m_k & (\forall n \leq k)[F \in A_{m_k,n}], \\ m_k + 1 & \text{otherwise.} \end{cases}$$

It is easy to see that the mapping $\Gamma = \lambda F.\, P_F$ is P-computable. Moreover, if $F \in X$ then $\lim_k m_k$ exists and equals the least m such that $F \in A_{m,n}$ for all n. If k_0 is such that $m_k = m$ for all $k \geq k_0$ then for all strings σ with $|\sigma| \geq k_0 + 1$ we have that $P_F(\sigma) = P(P_F[\sigma^-])$, hence P and P_F coordinate. Otherwise, there exist infinitely many k such that $m_{k+1} \neq m_k$, and for all strings σ with $|\sigma| = k+1$ we have that $P_F(\sigma) \neq P(P_F[\sigma^-])$ hence P and P_F do not coordinate.

(ii). It is sufficient to prove that TSCOPE(P) intersects any basic open set, i.e. any function f from a finite subset E of $\{0,1\}^*$ into $\{0,1\}$ can be extended to a total player Q which coordinates with P.

Given f, Q can be inductively defined as follows: suppose that we have defined $Q(\tau)$ for all $\tau \in \{0,1\}^*$ with $|\tau| < n$, and let $\sigma \in \{0,1\}^*$ with $|\sigma| = n$. Thus for all $i < n$ we can compute $R_{P,Q}(i)$ and consequently $R_{P,Q} \restriction n$. We define:

$$Q(\sigma) = \begin{cases} f(\sigma) & \text{if } \sigma \in E \\ P(Q[\sigma^-]) & \text{otherwise,} \end{cases}$$

(where we understand $Q[\sigma^-] = \emptyset$, if $\sigma = \emptyset$).

It is easily seen that Q extends f and coordinates with P.

(iii). It suffices to show that for every n the closed set $\bigcap_{m \geq n} A(P,m)$ does not contain any basic clopen set, where $A(P,m)$ is defined as in the proof of (i). In other words we aim to prove that any function f from a finite subset E of $\{0,1\}^*$ into $\{0,1\}$ can be extended to a total player Q such that $Q \notin \bigcap_{m \geq n} A(P,m)$. For this it suffices to define:

$$Q(\sigma) = \begin{cases} f(\sigma) & \text{if } \sigma \in E \\ 1 - P(Q[\sigma^-]) & \text{otherwise.} \end{cases}$$

(iv). Since the Lebesgue measure μ on $\{0,1\}^{\{0,1\}^*}$ is countably additive, it is sufficient to prove that for all $n \in \mathbb{N}$, $\bigcap_{m \geq n} A(P,m)$ has measure zero.

For $k \geq m$, let $B(P,m,k) = \bigcap_{m \leq i \leq k} A(P,i)$. Then $B(P,m,k) \cap A(P,k+1)$ and $B(P,m,k) \setminus A(P,k+1)$ have the same measure, therefore

$$\mu(B(P,m,k+1)) = \frac{1}{2} \cdot \mu(B(P,m,k)).$$

Thus $\mu(B(P,m,m+h)) = \frac{1}{2^h} \cdot \mu(A(P,m))$, and the claim follows.

(v). Let S be the set of all sequences σ such that for some $i < |\sigma|$, $\sigma(i) \neq R_{P,P}(i)$. Notice that $S \leq_T P$. Let ρ_i be the i-th string of S (in the order given by identification of $\{0,1\}^*$ with $\mathbb{N}$). Let $f : \mathbb{N} \longrightarrow \{0,1\}$ be any function with $P \leq_T f$. Define

$$P_f(\sigma) = \begin{cases} f(i) & \text{if } \sigma \in S,\ \sigma = \rho_i, \\ P(\sigma) & \text{if } \sigma \notin S. \end{cases}$$

Then $P_f \in \mathrm{TSCOPE}(P)$: In fact for every i, $R_{P,P_f}(i) = R_{P_f,P}(i)$. In particular $S \leq_T P_f$, as $R_{P,P}(i) = R_{P_f,P_f}(i)$ for every i. Then to compute $f(i)$ first find ρ_i (using an oracle for P_f) and then compute $P_f(\rho_i)$: thus $f \leq_T P_f$. On the other hand,

$$P_f \leq_T P \oplus f \equiv_T f.$$

It follows that $f \equiv_T P_f$. Thus TSCOPE(P) contains a player of the same Turing degree as f. ∎

We remind the reader that one can reformulate (i) of Theorem 23 by saying that TSCOPE(P) is F_σ-complete with respect to Wadge reducibility, see e.g. [15].

For the next corollary we recall that a set X is G_δ if it is a countable intersection of open sets, i.e. the complement of X is F_σ.

Corollary 24 *For every P,* $\mathrm{TSCOPE}(P) \in F_\sigma \setminus G_\delta$.

Proof. It is known, see e.g. [15], that there exists some $X \in F_\sigma \setminus G_\delta$ (in fact $X \in \Sigma^0_2 \setminus G_\delta$). Then there exists a continuous mapping Γ such that $X = \Gamma^{-1}(\mathrm{TSCOPE}(P))$. If TSCOPE($P$) were in G_δ then X would be in G_δ as well, a contradiction. ∎

In particular:

Corollary 25 *If P is computable then:*

(a) TSCOPE(P) *is a Σ^0_2 subset of $\{0,1\}^{\{0,1\}^*}$, and for every Σ^0_2 subset of $\{0,1\}^{\{0,1\}^*}$ there is a computable mapping $\Gamma : \{0,1\}^{\{0,1\}^*} \longrightarrow \{0,1\}^{\{0,1\}^*}$ such that*

$$X = \Gamma^{-1}(\mathrm{TSCOPE}(P)).$$

Hence $\mathrm{TSCOPE}(P) \in \Sigma^0_2 \setminus G_\delta$.

(b) TSCOPE(P) *intersects every Turing degree* $\mathbf{a}$.

As a side remark to Theorem 23 (v), we note that there are players P such that the spectrum of Turing degrees of players learned by P consists of exactly the Turing degrees above $\deg_T(P)$. In fact:

Corollary 26 *If P is a blind player then for every $Q \in$ TSCOPE(P) we have that $P \leq_T Q$.*

Proof. Suppose that P is a blind player, let $Q \in$ TSCOPE(P), and let n be such that for every $m \geq n$, $R_{P,Q}(m) = R_{Q,P}(m)$. Then for every σ such that $|\sigma| = m \geq n+1$ we have

$$P(\sigma) = P(R_{Q,P} \restriction m) = Q(R_{P,Q} \restriction m),$$

which easily implies that $P \leq_T Q$. ∎

Next, as mentioned in Section 1 above, we show a main result of this section, that the competencies, [14, Page 369], of any two total players are the same modulo isomorphism, that is,

Theorem 27 *If P and Q are arbitrary total players, then* TSCOPE(P) *and* TSCOPE(Q) *are P,Q-computably homeomorphic. In particular if P,Q are computable then* TSCOPE(P) *and* TSCOPE(Q) *are computably homeomorphic.*

Proof. We define, for any ordered pair (P,Q) of total players, a map $\lambda S.\, S^{PQ}$ on $\{0,1\}^{\{0,1\}^*}$, and we show that the map $\lambda S.\, S^{QP}$ associated to the pair (Q,P) is the inverse of $\lambda S.\, S^{PQ}$. Moreover, we prove that for every total player S, one has: $S \in$ TSCOPE(P) if and only if $S^{PQ} \in$ TSCOPE(Q). Finally, $\lambda S.\, S^{PQ}$ is P,Q-computable. Thus $\lambda S.\, S^{PQ}$ is a P,Q-computable homeomorphism of $\{0,1\}^{\{0,1\}^*}$ onto itself. This is clearly sufficient to prove Theorem 27.

First of all some notation: for every quadruple of (not necessarily total) players A,B,C,D defined on all strings τ with $|\tau| < n$ and for every nonempty string $\sigma \in \{0,1\}^*$ with $|\sigma| = n$ we define $\sigma^{ABCD} \in \{0,1\}^*$, with $|\sigma^{ABCD}| = |\sigma|$ as follows: For $i < |\sigma|$, let

$$\sigma^{ABCD}(i) = \begin{cases} R_{C,D}(i) & \text{if } \sigma(i) = R_{A,B}(i), \\ 1 - R_{C,D}(i) & \text{otherwise.} \end{cases}$$

Let S be any total player. We define a player S^{PQ} (computable in S,P,Q) by induction as follows:

First of all, we define:

$$S^{PQ}(\emptyset) = \begin{cases} Q(\emptyset) & \text{if } P(\emptyset) = S(\emptyset); \\ 1 - Q(\emptyset) & \text{otherwise.} \end{cases}$$

Now let $\sigma \in \{0,1\}^*$ with $|\sigma| = n > 0$, and assume that we have defined $S^{PQ}(\tau)$ for all $\tau \in \{0,1\}^*$ with $|\tau| < n$. Note that under this assumption we can compute (with an oracle for P, Q, S) $R_{Q,S^{PQ}}(i)$ and $R_{S^{PQ},Q}(i)$ for all $i < n$. Then we define $S^{PQ}(\sigma)$ by cases:

Case (a) For all $i < n$, $\sigma(i) = R_{Q,S^{PQ}}(i)$. Then:

(a1) If $S(\sigma^{QS^{PQ}PS}) = P(S[(\sigma^{QS^{PQ}PS})^-])$ then $S^{PQ}(\sigma) = Q(S^{PQ}[\sigma^-])$.
(a2) Otherwise, $S^{PQ}(\sigma) = 1 - Q(S^{PQ}[\sigma^-])$.

Case (b) There is $i < n$ such that $\sigma(i) \neq R_{Q,S^{PQ}}(i)$. Then let $S^{PQ}(\sigma) = S(\sigma^{QS^{PQ}PS})$.

This concludes the definition of S^{PQ}. Note that S^{PQ} is computable in S, P and Q, thus if P and Q are computable players, then S^{PQ} is computable in S, and the mapping $\lambda S.\, S^{PQ}$ is computable.

Moreover our procedure allows us to define $\lambda S.\, S^{PQ}$ for *all* pairs (P, Q) of total players, thus it makes sense to consider e.g. $(S^{AB})^{CD}$, (the result of applying to S the composition of the operators ^{AB} and ^{CD}) where S, A, B, C, D are arbitrary total players. The following claim is immediate from the definitions of S^{PQ} and $(S^{PQ})^{QP}$:

Claim A. Let $\delta \in \{0,1\}^*$ with $|\delta| > 0$ and let $\gamma = \delta^{P(S^{PQ})^{QP}QS^{PQ}}$. Then for $i < |\delta|$, the following are equivalent:

(i) $\delta(i) = R_{P,(S^{PQ})^{QP}}(i)$.
(ii) $\gamma(i) = R_{Q,S^{PQ}}(i)$.
(iii) $\gamma^{Q(S^{PQ})PS}(i) = R_{P,S}(i)$.

Next we prove the following claim:

Claim B. For all total players P, Q and S, and for all $\rho \in \{0,1\}^*$, $S(\rho) = (S^{PQ})^{QP}(\rho)$.

Proof of Claim B. We start with the case $\rho = \emptyset$. If $\rho = \emptyset$, then $S(\rho) = P(\rho)$ if and only if $S^{PQ}(\rho) = Q(\rho)$, if and only if $(S^{PQ})^{QP}(\rho) = P(\rho)$, therefore $S(\rho) = (S^{PQ})^{QP}(\rho)$.

Suppose that $|\rho| = n > 0$, and that the claim holds for all $\tau \in \{0,1\}^*$ such that $|\tau| < n$. Then for all $i < n$, $R_{P,S}(i) = R_{P,(S^{PQ})^{QP}}(i)$, and $R_{S,P}(i) = R_{(S^{PQ})^{QP},P}(i)$. Therefore, letting $\alpha = \rho^{P(S^{PQ})^{QP}QS^{PQ}}$, by Claim A we get: $\rho = \alpha^{QS^{PQ}PS}$.

If $(S^{PQ})^{QP}(\rho)$ is defined according to Case (a), then by Claim A, $S^{PQ}(\alpha)$ is also defined according to Case (a), therefore:

(a1) If $(S^{PQ})^{QP}(\rho)$ is defined through case (a1), then an easy calculation shows that $S^{PQ}(\alpha) = Q(S^{PQ}[\alpha^-])$, i.e. $S^{PQ}(\alpha)$ is defined through (a1) as well. Then using the Induction Hypothesis,

$$\begin{aligned} S(\alpha^{QS^{PQ}PS}) &= S(\rho) \stackrel{\text{(a1)}}{=} P(S[\rho^-]) \stackrel{\text{IH}}{=} P((S^{PQ})^{QP}[\rho^-]) \\ &= P((S^{PQ})^{QP}[(\alpha^{QS^{PQ}PS})^-]) \\ &\stackrel{\text{(a1)}}{=} (S^{PQ})^{QP}(\alpha^{QS^{PQ}PS}) \\ &= (S^{PQ})^{QP}(\rho). \end{aligned}$$

(The equality $\stackrel{\text{(a1)}}{=}$ is justified by the definition of S^{PQ} in clause (a1). Again, the equality $\stackrel{\text{IH}}{=}$ is justified by the Induction Hypothesis.)

(a2) As before we can argue that if $(S^{PQ})^{QP}(\rho)$ is defined through case (a2), then $S^{PQ}(\alpha) \neq Q(S^{PQ}[\alpha^-])$, i.e. $S^{PQ}(\alpha)$ is defined through (a2) as well. Then

$$\begin{aligned} S(\alpha^{QS^{PQ}PS}) &= S(\rho) =^{\text{(a2)}} 1 - P(S[\rho^-]) \\ &\stackrel{\text{IH}}{=} 1 - P((S^{PQ})^{QP}[(\alpha^{QS^{PQ}PS})^-]) \\ &\stackrel{\text{(a2)}}{=} (S^{PQ})^{QP}(\alpha^{QS^{PQ}PS}) \\ &= (S^{PQ})^{QP}(\rho). \end{aligned}$$

(As before, the equality $\stackrel{\text{(a2)}}{=}$ is justified by the definition of S^{PQ} in clause (a2).)

Now suppose that $(S^{PQ})^{QP}(\rho)$ is defined according to Case (b). Then, by Claim A, $S^{PQ}(\alpha)$ is also defined according to Case (b). Thus

$$(S^{PQ})^{QP}(\rho) = S^{PQ}(\alpha) = S(\alpha^{QS^{PQ}PS}) = S(\rho),$$

and Claim B is proved.

We conclude the proof of Theorem 27. By Claim B, for all total players P and Q, the map $\lambda S.S^{PQ}$ is a bijection of $\{0,1\}^{\{0,1\}^*}$ onto itself, because $\lambda S.S^{QP}$ inverts $\lambda S.S^{PQ}$ and viceversa. Clearly $\lambda S.S^{PQ}$ is P,Q-computable, because for all $\tau \in \{0,1\}^*$, and for every total player S, $S^{PQ}(\tau)$ only depends on the values of S on a finite subset of $\{0,1\}^*$ (namely the set of all $\delta \in \{0,1\}^*$ with $|\delta| < |\tau|$).

In order to conclude the proof of Theorem 27, it remains to prove that $S \in \text{TSCOPE}(P)$ if and only if $S^{PQ} \in \text{TSCOPE}(Q)$. Now the definition of S^{PQ}, Case (a), makes sure that for all $i \in \mathbb{N}$, $R_{S,P}(i) = R_{P,S}(i)$ if and only if $R_{S^{PQ},Q}(i) = R_{Q,S^{PQ}}(i)$. This concludes the proof. ∎

In the rest of this section we give a completely *effective version* of the competency isomorphism result above (Theorem 27). Specifically, we show that, if P, Q are total computable players then the classes of total computable players learned by P and Q, respectively, have computably isomorphic index sets. Throughout the rest of the section, the symbols Σ_n^0 and Π_n^0 refer to levels of the arithmetical hierarchy of sets of natural numbers, see e.g. [21, Chapter IV].

Given a player P define $\text{IND}(P)$ to be the index set of the class of total computable players learned by P, i.e.,

$$\text{IND}(P) = \{n : \varphi_n \text{ total and } \varphi_n \text{ is learned by } P\}.$$

Let $\text{Tot} = \{n : \varphi_n \text{ total}\}$. It is well known (see e.g. [21, Page 66]) that Tot is a Π_2^0 set, in fact Π_2^0-complete. Now,

$$n \in \text{IND}(P) \Leftrightarrow n \in \text{Tot and } (\exists k_0)(\forall k \geq k_0)[R_{P,\varphi_n}(k) = R_{\varphi_n,P}(k)].$$

Thus if P is computable then an easy calculation shows that $\text{IND}(P)$ is Δ_3^0. We can be more precise about the arithmetical complexity of $\text{IND}(P)$. Let us say that a subset $X \subseteq \mathbb{N}$ is 2-Σ_2^0 if $X = Y \setminus Z$, where $Y, Z \in \Sigma_2^0$. (The 2-Σ_2^0 sets constitute one of the levels of the Ershov difference hierarchy of the Σ_2^0 sets. For more on the Ershov hierarchy, see e.g. [6]).

What we have seen above clearly amounts to the following:

Lemma 28 *If P is computable then* $\text{IND}(P)$ *is* 2-Σ_2^0.

Theorem 29 *For every total computable player P, the set* $\text{IND}(P)$ *is* 2-Σ_2^0-*complete, i.e. for every* 2-Σ_2^0 *set X one has* $X \leq_1 \text{IND}(P)$.

Proof. Let $X = Y \cap Z$ where $X \in \Sigma_2^0$ and $Z \in \Pi_2^0$, and let $R_Y(x,t,s)$, $R_Z(x,t,s)$ be computable relations such that

$$Y = \{i : (\exists t)(\forall s)R_Y(i,t,s)\} \qquad Z = \{i : (\forall t)(\exists s)R_Z(i,t,s)\}.$$

Moreover let P be any total computable player.

By the s_n^m-Theorem let g be a $1-1$ computable function such that:

- $\varphi_{g(i)}(\emptyset) = 0$; let also $m_0 = 0$;
- Suppose $|\sigma| = k+1$ and we have inductively defined m_k and $\varphi_{g(i)}(\tau)$ for all τ such that $|\tau| \leq k$. Then let $\varphi_{g(i)}(\sigma) =\uparrow$, if $\varphi_{g(i)}(\tau) =\uparrow$ for some τ with

$|\tau| \leq k$; otherwise let:

$$\varphi_{g(i)}(\sigma) = \begin{cases} \uparrow & \text{if } (\forall s)[\neg R_Z(i,k,s)], \\ P(\varphi_{g(i)}[\sigma^-]) & \text{if } (\exists s)[R_Z(i,k,s)] \text{ and} \\ & (\forall s \leq k)[R_Y(i,m_k,s)] \\ 1 - P(\varphi_{g(i)}[\sigma^-]) & \text{otherwise.} \end{cases}$$

Moreover, let

$$m_{k+1} = \begin{cases} m_k & \text{if } (\forall s \leq k)[R_Y(i,m_k,s)], \\ m_k + 1 & \text{otherwise.} \end{cases}$$

To verify that for all i

$$i \in X \Leftrightarrow g(i) \in \text{IND}(P)$$

first note that if $i \notin Z$, then there exists some k such that, for no s do we have $R_Z(i,k,s)$; hence $g(i) \notin \text{IND}(P)$ as $\varphi_{g(i)}$ is not total. Next suppose that $i \in Z$. In this case $\varphi_{g(i)}$ is total. If $i \in Y$ then $m = \lim_k m_k$ exists, and m is the least number such that $R_Y(i,m,s)$ for all s. If k_0 is such that $m_k = m$ for all $k \geq k_0$, then for all strings σ with $|\sigma| \geq k_0$ we have that $\varphi_{g(i)}(\sigma) = P(\varphi_{g(i)}[\sigma^-])$; hence $\varphi_{g(i)}(\sigma)$ is learned by P. If $i \notin Y$ then there are infinitely many k such that $m_{k+1} \neq m_k$, and for all strings σ with $|\sigma| = k+1$ we have that $\varphi_{g(i)}(\sigma) \neq P(\varphi_{g(i)}[\sigma^-])$; hence $\varphi_{g(i)}$ and P do not coordinate. ▮

Next is the completely effective competency isomorphism result mentioned in Section 1 as one of the main results of the present section.

Corollary 30 *If P and Q are total computable players then* IND(P) *and* IND(Q) *are computably isomorphic.*

Proof. Immediate by the Myhill Isomorphism Theorem (see e.g. [20, Page 85]) as by the previous theorem we have $\text{IND}(P) \leq_1 \text{IND}(Q)$ and $\text{IND}(Q) \leq_1 \text{IND}(P)$. ▮

As to be expected, more powerful coordinators can learn more complicated sets.

Theorem 31 *There exists a total player $P \leq_T \emptyset'$ such that* IND(P) *is Σ^0_3-complete.*

Proof. First notice that if $P \leq_T \emptyset'$ then IND(P) is Σ^0_3 as for $n \in \text{Tot}$ the predicate $R_{P,\varphi_n}(k) = R_{\varphi_n,P}(k)$ is Δ^0_2. Thus we have only to show that there exists a player P such that $S \leq_m \text{IND}(P)$ for every $S \in \Sigma^0_3$.

We refer to the following:

Lemma 32 *If S is a Σ^0_3 set then there exists a uniformly computably enumerable sequence $\{X_{\langle i,n\rangle}\}_{i,n}$ of sets (meaning that the predicate $R(x,i,n) \Leftrightarrow^{\text{dfn}} x \in X_{\langle i,n\rangle}$ is recursively enumerable) such that*

$$i \in S \Leftrightarrow (\exists n)[X_{\langle i,n\rangle} \text{ infinite}].$$

Proof. See e.g. [21, Page 67]. ∎

We now continue with the proof of the theorem. Let S be a Σ^0_3-complete set, and let us fix a uniformly computable enumerable sequence $\{X_{\langle i,n\rangle}\}$ of sets as above. Let $\{X_{\langle i,n\rangle,s}\}_{i,n,s}$ be a uniformly computable sequence of finite sets such that for every i, n

$$X_{\langle i,n\rangle} = \bigcup_s X_{\langle i,n\rangle,s}.$$

By the s^m_n-Theorem let g be a $1-1$ computable function such that for every i, $\varphi_{g(i)}$ is defined as follows:

$$\varphi_{g(i)}(\sigma) = \begin{cases} 0 & \text{if } |\sigma| < i \\ 1 & \text{if } |\sigma| = i \\ j & \text{if otherwise and } \sigma = \sigma^- \cdot j. \end{cases}$$

We now define P. Let us say that a string σ is i, n*-regular* if $0^i1 \subseteq \sigma$, and the string 11 is not a substring of σ, and σ contains exactly $n+1$ occurrences of the bit 1. Suppose that we have already defined $P(\tau)$ for every τ such that $|\tau| < |\sigma|$: We define $P(\sigma)$ and $P(\sigma \cdot j)$ for all $j \in \{0,1\}$ as follows:

$$P(\sigma) = \begin{cases} 0 & \text{if } \sigma \text{ is } i,n\text{-regular for some (necessarily unique) } i,n \\ & \text{and } (\exists s > |\sigma|)[X_{\langle i,n\rangle,s} \setminus X_{\langle i,n\rangle,|\sigma|} \neq \emptyset], \\ 1 & \text{if } \sigma \text{ is } i,n\text{-regular for some (necessarily unique) } i,n \\ & \text{and } (\forall s > |\sigma|)[X_{\langle i,n\rangle,s} = X_{\langle i,n\rangle,|\sigma|} = \emptyset], \\ 0 & \text{otherwise.} \end{cases}$$

Next, for every $j \in \{0,1\}$ define

$$P(\sigma \cdot j) = 0.$$

Assume that $i \in S$, and let n be the least number such that $X_{\langle i,n\rangle}$ is infinite. Then $\varphi_{g(i)}$ and P play respectively infinite strings of the form

$$R_{\varphi_{g(i)},P} = 0^i10^{h_0}010^{h_1}010^{h_2}\dots010^{h_{n-1}}010^{\infty}$$
$$R_{P,\varphi_{g(i)}} = 0^i00^{h_0}100^{h_1}100^{h_2}\dots100^{h_{n-1}}10^{\infty}$$

thus P and $\varphi_{g(i)}$ coordinate. (For every $k < n$, the number h_k depends on the cardinality of the *finite* set $X_{\langle i,k\rangle}$. Eventually P always plays 0 because $X_{\langle i,n\rangle}$ is infinite.)

Otherwise, $\varphi_{g(i)}$ and P play respectively infinite strings of the form

$$R_{\varphi_{g(i)},P} = 0^i 1 0^{h_0} 010^{h_1} 010^{h_2} 01 \dots 010^{h_r} 01 \dots$$
$$R_{P,\varphi_{g(i)}} = 0^i 0 0^{h_0} 100^{h_1} 100^{h_2} 10 \dots 100^{h_r} 10 \dots$$

thus P and $\varphi_{g(i)}$ do not coordinate: For each i, following the block 0^{h_i}, $\varphi_{g(i)}$ always plays 0 and P plays 1.

Then for every i,

$$i \in S \Leftrightarrow g(i) \in \mathrm{IND}(P)$$

giving that $S \leq_1 \mathrm{IND}(P)$, hence $\mathrm{IND}(P)$ is Σ_3^0-complete. ∎

Acknowledgments. We would like to thank the referees for very helpful comments. Grant support was received by J. Case from NSF grant CCR-0208616 and by S. Jain from NUS grant R252-000-127-112.

References

[1] J. M. Barzdin. Prognostication of automata and functions. *Information Processing*, 1:81–84, 1971.

[2] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.

[3] J. Case, S. Jain, S. Kaufmann, A. Sharma, and F. Stephan. Predictive learning models for concept drift. *Theoretical Computer Science*, 268:323–349, 2001. Special Issue for *ALT'98*.

[4] J. Case, S. Jain, S. Lange, and T. Zeugmann. Incremental concept learning for bounded data mining. *Information and Computation*, 152:74–110, 1999.

[5] J. Case, M. Ott, A. Sharma, and F. Stephan. Learning to win process-control games watching game-masters. *Information and Computation*, 174(1):1–19, 2002.

[6] Yu. L. Ershov. A hierarchy of sets, I. *Algebra i Logika*, 7(1):47–74, January–February 1968. English Translation, Consultants Bureau, NY, pp. 25–43.

[7] R. Freivalds, E. Kinber, and C. Smith. On the impact of forgetting on learning machines. *Journal of the ACM*, 42:1146–1168, 1995.

[8] M. Fulk, S. Jain, and D. Osherson. Open problems in Systems That Learn. *Journal of Computer and System Sciences*, 49(3):589–604, December 1994.

[9] E.M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[10] S. Jain, D. Osherson, J. Royer, and A. Sharma. *Systems that Learn: An Introduction to Learning Theory.* MIT Press, Cambridge, Mass., second edition, 1999.

[11] M. Kummer and M. Ott. Learning branches and closed recursive games. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory,* Desenzano del Garda, Italy. ACM Press, July 1996.

[12] S. Kurtz, C. Smith and R. Wiehagen. On the role of search for learning from examples. *Journal of Experimental and Theoretical Artificial Intelligence*, 13:24–43, 2001.

[13] M. Minsky. *The Society of Mind.* Simon and Schuster, NY, 1986.

[14] F. Montagna and D. Osherson. Learning to coordinate: A recursion theoretic perspective. *Synthese*, 118:363–382, 1999.

[15] Y. N. Moschovakis. *Descriptive Set Theory.* North–Holland Publishing Co., Amsterdam, New York, Oxford, 1980.

[16] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, Cambridge, Mass., 1986.

[17] L. Pitt. *A Characterization of Probabilistic Inference.* PhD thesis, Yale University, 1984.

[18] L. Pitt. Probabilistic inductive inference. *Journal of the ACM*, 36:383–433, 1989.

[19] L. Pitt and C. Smith. Probability and plurality for aggregations of learning machines. *Information and Computation*, 77:77–92, 1988.

[20] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York, 1967.

[21] R. I. Soare. *Recursively Enumerable Sets and Degrees.* Perspectives in Mathematical Logic, Omega Series. Springer–Verlag, Heidelberg, 1987.