On Learning To Coordinate:
Random Bits Help, Insightful Normal Forms, and Competency Isomorphisms

John Case[1], Sanjay Jain[2], Franco Montagna[3], Giulia Simi[3], and Andrea Sorbi[3]

[1] Department of Computer and Information Sciences
University of Delaware, Newark, DE 19716-2586 (USA)
case@cis.udel.edu
[2] School of Computing
National University of Singapore, Singapore 117543, Republic of Singapore
sanjay@comp.nus.edu.sg
[3] Dipartimento di Scienze Matematiche ed Informatiche "R. Magari",
University of Siena, 53100 Siena, Italy
{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.

1 Introduction

The learning theory paradigms descending from Gold's [7], as summarized say in [8], do not generally take into account the delicacies of learning with 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 [9, 3] which study learning to win reactive process-control games and [1] 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 [11]. 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.*[4] Their goal, successful coordination, is to have their bit streams perfectly match past some point. [11] 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 [11] 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.

The main results of Section 3 concern probabilistic algorithmic coordinators **PM** surprisingly beating deterministic ones (Theorems 2, 3, 4, and 5) or beating probabilistic ones (Corollary 1) — 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 as they go — in memory limited senses identical to or similar to the senses studied in [13, Page 66] and $[6, 5, 11, 2]$.

Theorems 2 and 3 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 $\mathbf{M}_1, \ldots, \mathbf{M}_m$ so that for each element F of $\mathcal{C}$, at least i of $\mathbf{M}_1, \ldots, \mathbf{M}_m$ coordinate with F.

In the setting of learning programs for computable functions, team learning by deterministic machines is completely characterized by single machines learning probabilistically and vice versa [14–16]. 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

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

holds for learning to coordinate, and that Theorems 2 and 3 mentioned above are readily explainable as deterministic team learning in disguise. Theorems 4 and 5 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 [10]) for tasks such as muscle movement and speech, if random bits may need to be employed to achieve learning to coordinate.

Corollary 1 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 2 and 4 and Corollary 1 are additionally *blind*, i.e., they depend functionally only on the length of the bit strings they see. Theorems 3 and 5 are uniformizations of Theorems 2 and 4, respectively. More particularly, Theorems 3 and 5 feature the strong quantifier order $(\exists\mathcal{C})(\forall$ probabilities $p < 1)[\cdots]$; whereas, Theorems 2 and 4 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 3 and 5: the **PM**'s in the proofs of Theorems 3 and 5 are not blind, and, as we shall see, they are not as memory limited as the **PM**'s in the proofs of Theorems 2 and 4.[5]

[11, (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 [8]. In Section 4 we show (Theorem 8) that *any* total computable player P is (extensionally) equivalent to a player E_P based an enumeration strategy for a suitable indexed class of blind players. The enumeration strategy based E_P can and should be thought of 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 10 also in Section 4), but, of course, the point of the existence of E_P (and the "converse" result [11, (6), Page 367]) is the interesting insight that total coordinators can be *extensionally* conceptualized as exploiting an enumeration technique. From Theorem 10, 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 [11] it is shown that the competence, [11, p. 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 11 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 [12] (computably homeomorphic, if **P** and **Q** are computable). Then we provide a completely *effective version* of this competency isomorphism theorem. Let IND(P) denotes the index set of

[5] It is open whether these costs are necessary.

the restriction of TSCOPE(P) to computable players for a computable coordinator P. Corollary 2 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 2^* the set of all (finite) binary strings. If $\sigma \in 2^*$ then $|\sigma|$ denotes the length of σ; $\sigma(i)$, with $i < |\sigma|$, is the i-th bit of σ; $\sigma \restriction n$ denotes the initial segment of σ of length equal to n; (clearly $\sigma \restriction 0$ is the empty string, denoted by $\emptyset$); if $|\sigma| > 0$ then $\sigma^- = \sigma \restriction |\sigma| - 1$. The concatenation of two strings σ and τ is denoted by $\sigma \cdot \tau$. The symbol $\langle i_0, \dots, i_n \rangle$ denotes the string consisting of the bits $i_0, \dots, i_n$; if $n = 0$ we simply write i_0 for $\langle i_0 \rangle$.

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

A *player* (synonym: a *coordinator*) is a partial function from 2^* 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

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

where $R_{F,G}(n)$ is defined if and only if for every $i < n$, $R_{G,F}(i)$ is defined, and $F(R_{G,F} \restriction n)$ 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. 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 2^* into 2^*. Indeed, any player F originates a function from 2^* into 2^* as follows: F maps a string σ into a string $F[\sigma]$ with $|F[\sigma]| = |\sigma| + 1$, and for every $i < |\sigma| + 1$, $F[\sigma](i) = F(\sigma \restriction i)$ if $F[\sigma](j)$ converges for all $j < i$ and $F(\sigma \restriction i)$ converges; divergent otherwise. $F[\sigma]$ denotes the sequence τ (of length $|\sigma| + 1$) that F outputs while coordinating with some player which starts with σ when coordinating with F.

Conversely, every monotonic partial function $h : 2^* \longrightarrow 2^*$ 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 $\{\mathbf{M}_i\}_{i \in \mathbb{N}}$ of all (partial) computable deterministic players, and to some effective listing $\{\mathbf{PM}_i\}_{i \in \mathbb{N}}$ 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.

The first result in this section is just a lift of a result from [11].

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

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

Proof. This can be handled by straightforwardly generalizing the proof for the $n = 2$ case in [11]. ∎

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

Definition 1. [11] A player Q is *blind* if $Q(\sigma) = Q(\tau)$ for all $\sigma, \tau \in 2^*$ such that $|\sigma| = |\tau|$.

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

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

Theorem 2. *Suppose $0 \leq p < 1$. There exists a class of deterministic players $\mathcal{C}$ such that*

1. *No deterministic coordinator can coordinate with all of $\mathcal{C}$; and*
2. *For k chosen large enough that $1 - 2^{-k} \geq p$, there exists a* blind *probabilistic coordinator* **PM**, *such that:*
   (a) *For each member $\mathbf{M}$ of $\mathcal{C}$, $\mathbf{PM}$ can coordinate with $\mathbf{M}$ with probability $1 - 2^{-k} \geq p$; and*
   (b) *$\mathbf{PM}$ is k-memory limited in the sense of [13, Page 66]; more specifically, $\mathbf{PM}$ needs to remember whether it is outputting one of its first k bits — which are its only random bits.*

Proof. As above, let k be chosen large enough that $1 - 2^{-k} \geq p$. Let $\mathbf{M}_n$ be the n-th deterministic player. Define player F_n as follows. Let τ be any sequence (over 0, 1). $F_n[\tau]$ is defined as follows.

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

Note that F_n does not coordinate with $\mathbf{M}_n$. Thus no deterministic machine can coordinate with $\mathcal{C} = \{F_i \mid i \in \mathbb{N}\}$.

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

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

Theorem 3. *There exists a class of deterministic players $\mathcal{C}$ such that:*

1. *No deterministic coordinator can coordinate with all of $\mathcal{C}$; and*
2. *For all p such that $0 \leq p < 1$, for k chosen large enough that $1 - 2^{-k} \geq p$, there exists a probabilistic coordinator* **PM** *such that:*
   (a) *For each member* **M** *of $\mathcal{C}$,* **PM** *can coordinate with* **M** *with probability $1 - 2^{-k} \geq p$; and*
   (b) **PM** *is k-memory limited but only beyond the first 0 it sees, it needs to remember how many 1's it saw before the first 0, and, after the first 0 it sees, it outputs either a deterministic bit or k random bits and no more random bits.*

Proof. Let $\mathbf{M}_n$ be the n-th deterministic player. Define player F_n as follows. Let τ be any sequence (over 0, 1). $F_n[\tau]$ is defined as follows.

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

Note that F_n does not coordinate with $\mathbf{M}_n$. Thus no deterministic machine 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 $\mathbf{M}_i[1^i0] \neq 1^{i+1}b_i$. Now define **PM** as follows.

$$\mathbf{PM}[\tau] = \begin{cases} 1^{i+1}b_i1^{|\tau|-i-1}, \text{if } \tau \text{ starts with } 1^i0 \text{ for some } i \leq k; \\ 1^{i+1}[k \text{ random bits}]1^{|\tau|-k-i}, \text{if } \tau \text{ starts with } 1^i0 \text{ for some } i > k; \\ 1^{|\tau|+1}, \text{not above cases.} \end{cases}$$

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 $\mathbf{PM}[1^i01^i]$ same as $\mathbf{M}_i[1^i01^i]$. ∎

Theorem 4. *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 deterministic players $\mathcal{C}$ such that:

1. *$\mathcal{C} \notin \mathbf{Team}^1_m\mathbf{Coord}$;*
2. *There are $m+1$ blind deterministic machines such that:*
   (a) *these machine witness that $\mathcal{C} \in \mathbf{Team}^1_{m+1}\mathbf{Coord}$; and*
   (b) *each of these machines is $\lceil \log_2(m+1) \rceil$ memory limited; more specifically, each machine needs to remember whether it is outputting one of its first $\lceil \log_2(m+1) \rceil$ bits; and*
3. *There exists a probabilistic coordinator $\mathbf{PM}$, such that:*
   (a) *For each member $\mathbf{M}$ of $\mathcal{C}$, $\mathbf{PM}$ can coordinate with $\mathbf{M}$ with probability $\frac{2^k - m}{2^k} \geq p$; and*
   (b) *$\mathbf{PM}$ is k-memory limited; more specifically, $\mathbf{PM}$ needs to remember whether it is outputting one of its first k bits — which are its only random bits.*

Theorem 5. *Suppose $m \in \mathbb{N}^+$. Then there exists a class of deterministic players $\mathcal{C}$ such that:*

1. *$\mathcal{C} \notin \mathbf{Team}^1_m\mathbf{Coord}$;*
2. *There are $m+1$ blind deterministic machines such that:*
   (a) *these machine witness that $\mathcal{C} \in \mathbf{Team}^1_{m+1}\mathbf{Coord}$; and*
   (b) *each of these machines is $\lceil \log_2(m+1) \rceil$ memory limited; more specifically, each machine needs to remember whether it is outputting one of its first $\lceil \log_2(m+1) \rceil$ bits; and*
3. *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 probabilistic coordinator $\mathbf{PM}$ such that:*
   (a) *For each member $\mathbf{M}$ of $\mathcal{C}$, $\mathbf{PM}$ can coordinate with $\mathbf{M}$ with probability $\frac{2^k - m}{2^k} \geq p$; and*
   (b) *$\mathbf{PM}$ is k-memory limited but only beyond the first 0 it sees, it needs to remember how many 1's it saw before the first 0, and, after the first 0 it sees, it outputs either a deterministic bit or k random bits and no more random bits.*

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 6. *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 deterministic players $\in (\mathbf{Team}^\ell_m\mathbf{Coord} - \mathbf{Prob}_p\mathbf{Coord})$.*

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

The next corollary is one of the main results of this section mentioned in Section 1 above.

Corollary 1. *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 such that $k = \lceil \log_2(m) \rceil$, 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 7. *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 $\mathbf{M}_i$ 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 $\mathrm{card}(\{j \mid 1 \leq j \leq w$ and $\mathbf{M}_{\pi^w_j(n)}[1^n01^k]$ halts within t steps, and starts with $\sigma \in Y\}) < v$.

Define $F_n[\tau]$ as follows:

- if $r < n+k$ then let

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

- if $r \geq n+k$ 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 $\mathbf{M}_{\pi^w_j(n)}[F_n[\tau \restriction r-1]]\downarrow \supseteq \tau \restriction r \cdot x$ 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 $S^t_n = S_n$, 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 $\mathbf{M}_{\pi^w_1(n)}, \ldots, \mathbf{M}_{\pi^w_w(n)}$, which starts with σ while interacting with F_n, can coordinate with F_n.

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

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

Theorem follows. ∎

Proposition 1. *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 $\mathbf{M}^1, \ldots, \mathbf{M}^m$ are given. Define $\mathbf{M}^{',1}, \ldots, \mathbf{M}^{',w}$ such that $\mathbf{M}^{',i}$ follows $\mathbf{M}^j$, 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 $\mathbf{M}^1, \ldots, \mathbf{M}^m$ are also $\mathbf{Team}_w^v\mathbf{Coord}$ identified by $\mathbf{M}^{',1}, \ldots, \mathbf{M}^{',w}$. ∎

4 Normal form characterization of learnable classes of computable players

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

Definition 2. 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 3. Let $\mathcal{C}$ be an indexed, and dense class ("dense" means that for every $\sigma \in 2^*$ 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 2^* into $\{0,1\}$ as follows. Given $\sigma \in 2^*$, 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 [8] $\mathcal{C}$ 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 4. Let $\sigma \in 2^*$, and let P be a total computable player. We define a total computable function P_σ from $\mathbb{N}$ into $\{0,1\}$ as follows:

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

Definition 5. Let $\mathcal{C}$ and $C(i,x)$ be as in Definition 2. Define

$$\mathcal{P}(C) = \{P \mid R_{P,\mathrm{LE}(C)} = \mathrm{LE}(C)_\sigma\}$$

(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 $\sigma \in 2^*$. 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 8 next is a main result of this section mentioned above in Section 1. The player $\mathrm{LE}(C_P)$ of Theorem 8 can and should be thought of as an enumeration strategy based *canonical normal form of P*.

Theorem 8. *(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) $\mathcal{C}$ is a dense indexed class.
(ii) If $\sigma \subseteq \tau \subseteq P_\sigma$, then $P_\tau = P_\sigma$.
(iii) $\sigma_{i(\sigma)} \subseteq \sigma$, where $i(\sigma) = \min(\{i \in \mathbb{N} : (\forall j < |\sigma|)[C_P(i,j) = \sigma(j)]\})$.
(iv) $P_\sigma = P_{\sigma_{i(\sigma)}}$.
(v) $\mathrm{LE}(C_P) = P$.

We omit the proof of above claims. These claims imply part (a).

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

Definition 6. Given a player $\mathbf{P}$, we define:

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

Definition 7. An *indexed class of players* is a class $\mathcal{C}$ such that there is a total computable function $C(i, \sigma)$ from $\mathbb{N} \times 2^*$ 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 $\sigma \in 2^*$ $P(\sigma) = C(i, \sigma)$.

Theorem 9. *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 $Q_f(\sigma) = f(|\sigma|)$. It is easy to see that for every total computable player $\mathbf{P}$, BLINDSCOPE($\mathbf{P}$) $= B(\mathbf{P})$ where $B(\mathbf{P}) = \{Q_{\mathbf{P}_\sigma} : \sigma \in 2^*\}$. Hence a class of total computable blind players is learnable by a total computable player if and only if it is contained in $B(\mathbf{P})$ for some total computable player $\mathbf{P}$, and $B(\mathbf{P})$ is an indexed class of players, being enumerated by $C_P(i, \tau) = \mathbf{P}_{\sigma_i}(|\tau|)$. ∎

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 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 8 exploiting learning by enumeration for C_P. Clearly,

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

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 8 (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 10. *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

A total player can be regarded as an element of the topological space 2^{2^*} (with respect to the product topology, where 2 is equipped with the discrete topology). We recall that in this topological space the clopen sets form a basis: For every finite set $E \subseteq 2^*$ and every function $f \in 2^E$ let

$$C_f = \{F \in 2^{2^*} : (\forall \sigma \in E)[F(\sigma) = f(\sigma)]\};$$

then a nonempty set C is clopen if and only if there exists such a finite function f for which $C = C_f$. Given P, a mapping $\Gamma : 2^{2^*} \longrightarrow 2^{2^*}$ is called *P-computable* (or simply a *computable* if P is computable) if $\Gamma(F) = \lambda\sigma.\, \varphi_z^{F,P}(\sigma)$, for some index z (where $\varphi_z^{F,P}$ is the function computed by the Turing machine with index z and an oracle for F, P). It turns out that Γ is *continuous* if and only if it is P-computable for some P.

Next, as mentioned in Section 1 above, we show a main result of this section, that the competencies, [11, p. 369], of any two total players are the same modulo isomorphism, that is,

Theorem 11. *If* $\mathbf{P}$ *and* $\mathbf{Q}$ *are arbitrary total players, then* TSCOPE($\mathbf{P}$) *and* TSCOPE($\mathbf{Q}$) *are* $\mathbf{P}, \mathbf{Q}$*-computably homeomorphic. In particular if* $\mathbf{P}, \mathbf{Q}$ *are computable then* TSCOPE($\mathbf{P}$) *and* TSCOPE($\mathbf{Q}$) *are computably homeomorphic.*

Proof. We define, for any ordered pair $(\mathbf{P}, \mathbf{Q})$ of total players, a map $\lambda S.\, S^{\mathbf{PQ}}$ on 2^{2^*}, and we show that the map $\lambda S.\, S^{\mathbf{QP}}$ associated to the pair $(\mathbf{Q}, \mathbf{P})$ is the inverse of $\lambda S.\, S^{\mathbf{PQ}}$. Moreover, we prove that for every total player S, one has: $S \in$ TSCOPE($\mathbf{P}$) if and only if $S^{\mathbf{PQ}} \in$ TSCOPE($\mathbf{Q}$). Finally, $\lambda S.\, S^{\mathbf{PQ}}$ is $\mathbf{P}, \mathbf{Q}$-computable. Thus $\lambda S.\, S^{\mathbf{PQ}}$ is a $\mathbf{P}, \mathbf{Q}$-computable homeomorphism of 2^{2^*} onto itself. This is clearly sufficient to prove Theorem 11.

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 $\sigma \in 2^*$ with $|\sigma| = n$ we define $\sigma^{ABCD} \in 2^*$, 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 $S^{\mathbf{PQ}}$ (computable in $S, \mathbf{P}, \mathbf{Q}$) by induction as follows:

First of all, we define:

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

Now let $\sigma \in 2^*$ with $|\sigma| = n > 0$, and assume that we have defined $S^{\mathbf{PQ}}(\tau)$ for all $\tau \in 2^*$ with $|\tau| < n$. Note that under this assumption we can compute (with an oracle for $\mathbf{P}, \mathbf{Q}, S$) $R_{\mathbf{Q},S^{\mathbf{PQ}}}(i)$ and $R_{S^{\mathbf{PQ}},\mathbf{Q}}(i)$ for all $i < n$. Then we define $S^{\mathbf{PQ}}(\sigma)$ by cases:

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

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

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

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

Moreover our procedure allows us to define $\lambda S.\, S^{\mathbf{PQ}}$ for *all* pairs $(\mathbf{P}, \mathbf{Q})$ 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 rest of the proof relies on the following:

Claim. For all total players $\mathbf{P}, \mathbf{Q}$ and S, and for all $\rho \in 2^*$, $S(\rho) = (S^{\mathbf{PQ}})^{\mathbf{QP}}(\rho)$.

In order to prove the claim one can use the following fact: If $\delta \in 2^*$ has length $|\delta| > 0$ and $\gamma = \delta^{\mathbf{P}(S^{\mathbf{PQ}})^{\mathbf{QP}}\mathbf{Q}S^{\mathbf{PQ}}}$ then for every $i < |\delta|$, we have

$$\delta(i) = R_{\mathbf{P},(S^{\mathbf{PQ}})^{\mathbf{QP}}}(i) \Leftrightarrow \gamma(i) = R_{\mathbf{Q},S^{\mathbf{PQ}}}(i) \Leftrightarrow \gamma^{\mathbf{Q}(S^{\mathbf{PQ}})\mathbf{P}S}(i) = R_{\mathbf{P},S}(i).$$

We omit the proof of the claim. By Claim, for all total players $\mathbf{P}$ and $\mathbf{Q}$, the map $\lambda S.\, S^{\mathbf{PQ}}$ is a bijection, and is also $\mathbf{P}, \mathbf{Q}$-computable, because for all $\tau \in 2^*$, and for every total player S, $S^{\mathbf{PQ}}(\tau)$ only depends on the values of S on a finite subset of 2^*. Finally, it is straightforward to show that $S \in \mathrm{TSCOPE}(\mathbf{P})$ if and only if $S^{\mathbf{PQ}} \in \mathrm{TSCOPE}(\mathbf{Q})$. ∎

In the rest of this section we give a completely *effective version* of the competency isomorphism result above (Theorem 11). Specifically, we show that, if $\mathbf{P}, \mathbf{Q}$ are total computable players then the classes of total computable players learned by $\mathbf{P}$ and $\mathbf{Q}$, respectively, have computably isomorphic index sets.

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

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

Let $\mathrm{Tot} = \{n : \varphi_n \text{ total}\}$. It is well known (see e.g. [18, p. 66]) that Tot is a Π^0_2 set, in fact Π^0_2 complete. Now,

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

Thus if $\mathbf{P}$ is computable then an easy calculation shows that $\mathrm{IND}(\mathbf{P})$ is Δ^0_3. We can be more precise about the arithmetical complexity of $\mathrm{IND}(\mathbf{P})$. Let us say that a subset $X \subseteq \mathbb{N}$ is 2-Σ_2^0 if $X = Y - Z$, 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. [4]).

What we have seen above clearly amounts to the following:

Lemma 1. *If* $\mathbf{P}$ *is computable then* $\mathrm{IND}(\mathbf{P})$ *is* 2-Σ_2^0.

Theorem 12. *For every total computable player* $\mathbf{P}$, *the set* $\mathrm{IND}(\mathbf{P})$ *is* 2-Σ_2^0-*complete, i.e. for every* 2-Σ_2^0 *set* X *one has* $X \leq_1 \mathrm{IND}(\mathbf{P})$.

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 $\mathbf{P}$ 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 $m_0 = 0$;
- if $|\sigma| = k+1$ 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:

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

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

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

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 $g(i) \notin \mathrm{IND}(\mathbf{P})$ 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 $\varphi_{g(i)}(\sigma) = \mathbf{P}(\varphi_{g(i)}[\sigma^-])$; hence $\varphi_{g(i)}(\sigma)$ is learned by $\mathbf{P}$. 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 $\varphi_{g(i)}(\sigma) \neq \mathbf{P}(\varphi_{g(i)}[\sigma^-])$; hence $\varphi_{g(i)}$ and $\mathbf{P}$ 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 2. *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. [17, p. 85]) as by the previous theorem we have IND($\mathbf{P}$) $\leq_1$ IND($\mathbf{Q}$) and IND($\mathbf{Q}$) $\leq_1$ IND($\mathbf{P}$). ∎

Acknowledgments. 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. 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*.
2. J. Case, S. Jain, S. Lange, and T. Zeugmann. Incremental concept learning for bounded data mining. *Information and Computation*, 152:74–110, 1999.
3. 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.
4. 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.
5. R. Freivalds, E. Kinber, and C. Smith. On the impact of forgetting on learning machines. *Journal of the ACM*, 42:1146–1168, 1995.
6. 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.
7. E.M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
8. S. Jain, D. Osherson, J. Royer, and A. Sharma. *Systems that Learn: An Introduction to Learning Theory.* MIT Press, Cambridge, Mass., second edition, 1999.
9. 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.
10. M. Minsky. *The Society of Mind.* Simon and Schuster, NY, 1986.
11. F. Montagna and D. Osherson. Learning to coordinate: A recursion theoretic perspective. *Synthese*, 118:363–382, 1999.
12. Y. N. Moschovakis. *Descriptive Set Theory.* North–Holland Publishing Co., Amsterdam, New York, Oxford, 1980.
13. 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.
14. L. Pitt. *A Characterization of Probabilistic Inference.* PhD thesis, Yale University, 1984.
15. L. Pitt. Probabilistic inductive inference. *Journal of the ACM*, 36:383–433, 1989.
16. L. Pitt and C. Smith. Probability and plurality for aggregations of learning machines. *Information and Computation*, 77:77–92, 1988.
17. H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York, 1967.
18. R. I. Soare. *Recursively Enumerable Sets and Degrees.* Perspectives in Mathematical Logic, Omega Series. Springer–Verlag, Heidelberg, 1987.