Journal of Universal Computer Science, vol. 6, no. 8 (2000), 759-780
submitted: 16/7/99, accepted: 25/7/00, appeared: 28/8/00 © Springer Pub. Co.

Efficient Identification of Classes of P-Time Functions

Sandra Fontani
(University of Siena, ITALY
fontanis@mailsrv.unisi.it)

Abstract: We consider the problem of identifying a class of p-time functions in efficient time. We restrict our attention to particular classes of p-time functions, called uniform and we try to identify each function of such a class by guessing, after a small number of examples, some index for it or its next value. In both cases we introduce two efficient identification paradigms, called *efficient* and *very efficient identification* respectively. We find a characterization for efficient identification and, as a corollary, we show that the entire class P is not efficiently identifiable. A necessary condition is shown for very efficient identification, which becomes sufficient if and only if $\mathcal{P} = \mathcal{NP}$. We give some examples of well-known uniform classes which are very efficiently identifiable in both identification paradigms.
Key Words: Learning Theory.
Category: I.2.6.

1 Introduction

In this paper we are concerned with the following question: when can we consider an identification process of a class of functions to be successful in a reasonable (efficient) time? The main idea is to treat efficiency in terms of polynomiality, so it is natural to restrict our attention to the class of polynomial-time (p-time) functions, P. Informally, given a class of p-time functions together with a class of representations (*indexes*) for them, we will require the learner to be a p-time function and to become successful in a number of guesses polynomially bounded in the least index (or in the *length* of the least index) of the unknown function in the chosen representation class. This idea, as it stands, is rather vague because we have to state precisely which identification paradigms we refer to and what kind of classes of p-time functions it is reasonable to choose.

Regarding the first problem we will consider two identification paradigms, called EX and NV. They correspond respectively to the idea of identifying a function by constantly guessing, in the limit, an index for it (intuitively we "explain" the unknown function, because an index for it can be thought as a code for a program which computes it) or by correctly predicting, in the limit, its "next value". Clearly, if the learner has to guess an index for a p-time function, it is reasonable to require such an index to be relative to a particular indexing for P. Many definitions of indexing for P have been proposed [Buss 86]. We prefer to introduce a notion which is very similar to the standard definition of acceptable indexing for the class of partial recursive functions [Shoenfield 58].

Regarding the second problem, it is reasonable to require that, for every index h for a p-time function in a given class and sample S, the learner could check h to be consistent with S in polynomial time. This is certainly ensured if there is a polynomial-time algorithm which uniformly computes all the functions in the class (*universal function* for the class). Classes of p-time functions having such a property will be called *uniform*. We will see that the universal function chosen for a given uniform class, will allow us to associate to every p-time function in the class (at least) an index with respect to any particular acceptable indexing for P. So the class of representations for the functions of a uniform class will be completely specified by the particular universal function adopted for it.

We will define two efficient identification paradigms for uniform classes only, both in the EX and in the NV case. They will be distinguished by the size of the bound for the number of guesses allowed for the learner to become successful. More precisely, for every uniform class $\mathcal{C}$ with associated representation class $\mathcal{R}$, we say that $\mathcal{C}$ is *efficiently identifiable* if the learner becomes successful in a number of steps polynomially bounded by the least index of the unknown function in $\mathcal{R}$. We will say, instead, that $\mathcal{C}$ is *very efficiently identifiable*, if the learner becomes successful in a number of steps polynomially bounded by the *length* of the least index of the unknown function in $\mathcal{R}$. Since the length of each index is logarithmic in the least index itself, very efficient identification will imply efficient identification. Our paradigm shares with PAC learning [Kearns and Vazirani 94] the requirement about efficiency. Moreover, unlike PAC learning, we do not deal with probability nor with approximations: we only deal with exact, deterministic learning.

In Section 3 we formalize all concepts described above and we introduce the notions of *efficient* and *very efficient identification* for uniform classes in the EX and NV identification paradigms.

In Section 4 we exhibit some interesting examples of subclasses of P known in mathematics which turn out to be uniform and very efficiently identifiable.

In Section 5 we characterize efficient identification of uniform classes and we apply it to show that the entire class P is not efficiently identifiable.

In Section 6 we find a necessary condition for very efficient identification of uniform classes whose sufficiency turns to be equivalent to the well-known open problem $\mathcal{P} = \mathcal{NP}$.

2 Preliminaries

$\omega, \mathcal{Z}, \emptyset$ denote, respectively, the set of natural numbers, the set of integers and the emptyset. $\omega^{<\omega}$ and $2^{<\omega}$ denote the set of strings of natural numbers and the set of binary strings respectively. For every $a_0, ..., a_n \in \{0,1\}$, $a_0 a_1 ... a_n$ denotes the binary string whose elements are (in the given order) $a_0, a_1, ..., a_n$. In the same manner we interpret $(\tau_0, ..., \tau_n)$, if $\tau_0, ..., \tau_n \in \omega$ or $\tau_0, ..., \tau_n \in 2^{<\omega}$. We use symbols $f, g, h, \varphi, ..., \Psi, \Phi, \Gamma, \Delta, ...$ for recursive functions. We omit the arity of a function when it is clear from the context (or in the case of unary functions). We write $\lambda x_i.f(x_0, ..., x_n)$ $(i \leq n)$ to mean that f depends on variable x_i only (for every $j \leq n$, $j \neq i$, x_j is fixed).

We use p, q, p', q'... to indicate polynomials with positive integer coefficients. In the case of polynomials in one variable we sometimes omit the argument.

We write $min\, x[\cdots x \cdots]$ or $\mu x[\cdots x \cdots]$ (resp. $max\, x[\cdots x \cdots]$) to denote the least (resp. the greatest) natural number for which the expression $[\cdots x \cdots]$ is true when "x" is interpreted with such a value. If S is a set, $minS$ ($maxS$) denotes the least (the greatest) element of S and $card(S)$ indicates the cardinality of S.

2.1 Coding sequences

For every $a \in \omega$, we code a by its binary expansion, so, if $a = \sum_{i=0}^{n} \alpha_i 2^i$ $(n \in \omega$, $\alpha_i \in \{0,1\})$, where either $a = n = \alpha_0 = 0$ or $\alpha_n \neq 0$, we consider $a = \alpha_n \alpha_{n-1} ... \alpha_0$.

The length of a is the number of coefficients in its binary expansion and it is denoted by $|a|$. Clearly, for every $a \in \omega$, $|a| = \lceil log_2(a+1) \rceil$ (the least integer $\geq log_2(a+1)$), so we can approximate $|a|$ by $log_2(a)$. Note that $|0| = 0$.

The code for the numerical sequence $\overline{a} = (a_0, ..., a_n)$ is constructed as follows. We write the a_i's in binary notation, obtaining a string of 0, 1 and commas. We write such a string in reverse order. We replace each 0 by "10", each 1 by "11" and each comma by "00". The resulting string is the binary representation of the code of $\overline{a}$, which we denote by $< a_0, ..., a_n >$. For example, the code of $(3, 2, 4)$ is the number whose binary expansion is 101011001011001111. The code of (a) is the binary expansion of a and the code of the empty sequence is 0. We use the previous method to code finite sequences of binary strings too (not representing numbers at all).

Notice that, for every $(a_0, ..., a_n)$, $|< a_0, ..., a_n >| = 2(|a_0| + ... + |a_n| + n)$. Moreover there is a uniform effective method for checking if a number is the code of a finite sequence, which works in a number of steps polynomial in the length of the input sequence. We denote by Seq and $Bseq$ the set of codes of finite numerical sequences and the set of codes of finite sequences of binary strings respectively.

If finally Σ is a finite alphabet of symbols and Σ^* is the set of finite sequences of symbols in Σ, we can codify each $\sigma \in \Sigma^*$ in a similar manner. We associate to each $a \in \Sigma$ a number (different numbers for different symbols). The code of each $\sigma \in \Sigma^*$ is the code of the numerical sequence associated to it. Consider for example $\Sigma = \{x, |, \wedge, \vee, \_, (,)\}$. Each propositional formula A in conjunctive normal form ($A \in CNF$) can be represented as a sequence of symbols of Σ. Associate to each symbol in Σ the following numbers:

x	$\|$	$\wedge$	$\vee$	$\_$	(	)
1	2	3	4	5	6	7

Hence, if for example $A \equiv \overline{x}_1 \wedge (\overline{x}_2 \vee x_1)$, we write it as $A \equiv \_x| \wedge (\_x|| \vee x|)$ and the code of A is the code of $(5, 1, 2, 3, 6, 5, 1, 2, 2, 4, 1, 2, 7)$. We denote by $lth(A)$ the length of A in Σ and by $\lceil A \rceil$ the code of A. Notice that, for every $A \in CNF$, if $lth(A) = n$, then $|\lceil A \rceil| \leq 2(3n + n - 1) = 8n - 2$.

2.2 The class P

Throughout the paper we denote by P the class of functions which are computable in deterministic polynomial time (the definition of P is quite standard and can be found for example in [Buss 86]). Moreover we adopt the convention that all functions have domain ω^k and codomain ω. Without loss of generality, we consider as a model of computation deterministic Turing machines (T.m.). If M is a T.m. and $t : \omega^{n+1} \to \omega$ is any function, then, for all $a_0, ..., a_n \in \omega$, we write $M(a_0, ..., a_n) \downarrow \leq t(a_0, ..., a_n)$ if M on input $(a_0, ..., a_n)$ converges within $t(a_0, ..., a_n)$ steps of computation. In particular, M is deterministic polynomial time if and only if $M(a_0, ..., a_n) \downarrow \leq t(a_0, ..., a_n)$ where $t(a_0, ..., a_n) = p(|a_0|, ..., |a_n|)$ for some polynomial p. Sometimes, given a function $f : \omega^{n+1} \to \omega$ for which we have fixed a T.m. M that computes it, we write, by abuse of language, $f(a_0, ..., a_n) \downarrow \leq t(a_0, ..., a_n)$ instead of $M(a_0, ..., a_n) \downarrow \leq t(a_0, ..., a_n)$. We often refer to the following p-time functions:

(1) $\lfloor \frac{x}{2} \rfloor$: the integer part of $\frac{x}{2}$ (the greatest integer $\leq \frac{x}{2}$).

(2) $s(x) = x + 1$: the successor function.

(3)
$$x \dot{-} y = \begin{cases} x - y & \text{if } x \geq y \\ 0 & \text{otherwise} \end{cases}$$

(4)
$$\beta(i, < a_0, ..., a_n >) = \begin{cases} n + 1 & \text{if } i = 0 \\ a_{i-1} & \text{if } 0 < i \leq n + 1 \end{cases}$$

β is arbitrarily defined if $i > n + 1$ or if the second argument is not the code of any sequence [Buss 86]. For simplicity, we write $n+1$ for $\beta(0, < a_0, ..., a_n >)$ and, if $x \leq n$, we write a_x for $\beta(s(x), < a_0, ..., a_n >)$. Moreover, for every $n \in \omega$, we let $(n)_x = \beta(s(x), n)$ and $lth(n) = \beta(0, n)$. In particular, if $\sigma \in Seq$ ($\sigma \in Bseq$), say $\sigma = < a_0, ..., a_n >$, we have $(\sigma)_x = a_x$, $lth(\sigma) = n + 1$. We finally recall that P is closed under composition and limited iteration [Buss 86]. Informally, f is defined by limited iteration, if it is defined by recursion, but the number of steps of such recursion is logarithmic in the input and, at each step, the output of f is bounded by a (fixed) p-time function.

2.3 Identification paradigms

Consider some acceptable indexing $\pi_0, ..., \pi_n, ...$ for the class of partial recursive functions [Shoenfield 58]. We recall two very natural identification paradigms for classes of total recursive functions (see [Odifreddi 99] for a survey).

Definition 1 Let $\mathcal{C}$ be a class of total recursive functions. We say that $\mathcal{C}$ is *EX-identifiable* if there is a total recursive function g such that, for every $f \in \mathcal{C}$,

$$(\exists n_0)(\forall n \geq n_0) g(< f(0), ..., f(n) >) = i$$

for some $i \in \omega$ such that $f = \pi_i$.

Definition 2 Let $\mathcal{C}$ be a class of total recursive functions. We say that $\mathcal{C}$ is *NV-identifiable* if there is a total recursive function g such that, for every $f \in \mathcal{C}$,

$$(\exists n_0)(\forall n \geq n_0) g(< f(0), ..., f(n) >) = f(n+1).$$

3 Efficient identification setting

We define a particular indexing for the class P. As noted in the introduction, such an indexing will allow us to associate to each p-time function an index which, in some way, has the same "complexity" as the function itself. For instance, if f constantly assumes value n, then f may have an index bounded by a polynomial in n. It is in fact reasonable to require that to the extent that simple algorithms can be found for computing a p-time function, then correspondingly small indexes can be given to it.

Definition 3 An *acceptable indexing for P* is an enumeration $\varphi_0, ..., \varphi_n$ of (all of) p-time functions that meets the following conditions:

(i) There exist a function $\Phi(i,x)$ and a polynomial $p(i,x)$ such that, for every $i, x \in \omega$:

- $\Phi(i,x) \downarrow \leq p(i,x)$ $(\lambda ix.\Phi(|i|,|x|) \in P)$.
- $\Phi(i,x) = \varphi_i(x)$.

(ii) For every $\Psi(i,x) \in P$ there exists $h \in P$ strictly increasing such that, for every $i, x \in \omega$:

$$\Psi(i,x) = \Phi(h(i),x) = \varphi_{h(i)}(x).$$

Such a function $\Phi(i,x)$ will be called a *universal function for P*.

Theorem 1 *There exists an acceptable indexing for P, i.e. there exists a function $\Delta(i,x)$ satisfying the following conditions:*

(i) $\lambda ix.\Delta(|i|,|x|)\in P$.

(ii) For every $i\in\omega$, $\lambda x.\Delta(i,x)\in P$.

(iii) For every $f\in P$ there exists $i\in\omega$ such that $f=\lambda x.\Delta(i,x)$.

(iv) For every $\Psi(i,x)\in P$ there exists $h\in P$ strictly increasing such that, for every $i,x\in\omega$, $\Psi(i,x)=\Delta(h(i),x)$.

The proof of Theorem 1 is based on the following technical result:

Proposition 1 *There exists a function $\Gamma(i,x)$ such that:*

(i) $\lambda ix.\Gamma(|i|,|x|)\in P$.

(ii) For every $i\in\omega$, $\lambda x.\Gamma(i,x)\in P$.

(iii) For every $f\in P$ there exists $i\in\omega$ such that $f=\lambda x.\Gamma(i,x)$.

Proof The proof of Proposition 1 is quite standard, so we only sketch it. We only define the required function $\Gamma(i,x)$, leaving to the reader the easy proof of (i)–(iii). Let $F(i,x,k)$ be the output of the Turing machine M_i on input x after k steps of computation, if M_i on input x halts within k steps of computation, 0 otherwise. It is obvious that $F(i,x,k)$ converges in time linear in k. For every $s\in Seq$, for every $i,x,k\in\omega$ let:

$$G(i,s,x,k)=\begin{cases}(s)_x & \text{if } x<lth(s)\\ F(i,x,k) & \text{otherwise}\end{cases}$$

$$\Gamma(i,x)=G((i)_1,(i)_2,x,min\{|x+2|^i,x\}).$$

It is readily seen that Γ meets all requirements of Proposition 1. As for (iii), if f is in P, let e,k be such that M_e is a T.m. computing f and $M_e(x)\downarrow\leq|x+2|^k$ (k depends on the polynomial giving a bound to the steps of computation of M_e). Without loss of generality, we can suppose that $M_e(x)\downarrow\leq|x+2|^e$ (e.g. adding to the program for M_e a certain number of "dummy" instructions which don't affect the time of computation). Since e is fixed, from a certain x on, $|x+2|^e<x$. If n is the least of such x, letting $s=<f(0),...,f(n)>$ and $i=<e,s>$, it is easy to verify that $f=\lambda x.\Gamma(i,x)$.

Proof of Theorem 1 Let $\Gamma(i,x)$ be the function defined in the proof of Proposition 1. Let for every $i,x\in\omega$, $\Delta(i,x)=\Gamma((i)_1,<(i)_2,x>)$.

(i) By Proposition 1(i), there exists a polynomial t(i,x) such that, for all $i,x\in\omega$, $\Gamma(i,x)\downarrow\leq t(i,x)$. Moreover there exist polynomials $q(i)$ and $r(i,x)$ such that $(i)_1\downarrow\leq q(i)$ and $<(i)_2,x>\downarrow\leq r(i,x)$. Hence, letting $p(i,x)=t(q(i),r(i,x))$, $\Delta(i,x)\downarrow\leq p(i,x)$ (warning: Δ works in time polynomial in i,x, but not in $|i|$, $|x|$).

(ii) By Proposition 1(ii), for all $i\in\omega$, there exists a polynomial $q(x)$ (depending on $(i)_1$), such that, for all $x\in\omega$, $\Gamma(i,x)\downarrow\leq q(|x|)$. Moreover $|<(i)_2,x>|\leq 2(|i|+|x|+1)$. Hence, letting $p(x)=q(2(|i|+|x|+1))$, $\Delta(i,x)\downarrow\leq p(|x|)$.

(iii) Let $f\in P$ and, for every $x\in\omega$, $g(x)=f((x)_2)$. Clearly $g\in P$, so there exists $e\in\omega$ such that $g=\lambda x.\Gamma(e,x)$ (Proposition 1(iii)). Then:

$$\Delta(<e,e>,x)=\Gamma((<e,e>)_1,<(<e,e>)_2,x>)=\Gamma(e,<e,x>)=$$
$$g(<e,x>)=f((<e,x>)_2)=f(x).$$

If $i=<e,e>$, it follows that $f=\lambda x.\Delta(i,x)$.

(iv) Let $\Psi(i,x)\in P$ and let, for every $x\in\omega$, $k(x)=\Psi((x)_1,(x)_2)$. Clearly $k\in P$, hence there is $e\in\omega$ such that $k=\lambda x.\Gamma(e,x)$ (Proposition 1(iii)). For all $i,x\in\omega$:

$$\Delta(<e,i>,x)=\Gamma((<e,i>)_1,<(<e,i>)_2,x>)=$$
$$\Gamma(e,<i,x>)=k(<i,x>)=\Psi((<i,x>)_1,(<i,x>)_2)=\Psi(i,x).$$

So, if we let $h(i)=<e,i>$, $h\in P$ and $\lambda ix.\Psi(i,x)=\lambda ix.\Delta(h(i),x)$. Moreover h is strictly increasing since, for every $i,j\in\omega$, if $i<j$ then $<e,i><<e,j>$.

q.e.d.

We now introduce classes of p-time functions which are computable in a uniform and "efficient" way.

Definition 4 Let $\mathcal{C}\subseteq P$. We say that $\mathcal{C}$ is *uniform* if there exists $\Psi(i,x)$ such that:

- (i) $\lambda ix.\Psi(i,x)\in P$.
- (ii) For every $i\in\omega$, $\lambda x.\Psi(i,x)\in\mathcal{C}$.
- (iii) For every $f\in\mathcal{C}$, there exists $i\in\omega$ such that $f=\lambda x.\Psi(i,x)$.

Such a function $\Psi(i,x)$ is called a *universal function for* $\mathcal{C}$.

Notation 1 (1) By Definition 3, if $\mathcal{C}$ is a uniform class with universal function $\Psi(i,x)$, then there exists $h\in P$ strictly increasing such that, for every $i,x\in\omega$, $\Psi(i,x)=\varphi_{h(i)}(x)$. We call h a Ψ-indexing for $\mathcal{C}$ and we write:

$$\mathcal{C}=\mathcal{L}_h=\{\varphi_{h(i)}:i\in\omega\}.$$

(2) Let $\mathcal{L}_h=\{\varphi_{h(i)}:i\in\omega\}$ be a uniform class. For every $i\in\omega$ we define:

$$m(i)=min\{j\in\omega:\varphi_{h(j)}=\varphi_{h(i)}\}.$$

In other words, for every $\varphi_{h(i)}\in\mathcal{L}_h$, $\varphi_{h(i)}=\varphi_{h(m(i))}$ and the first occurrence of $\varphi_{h(i)}$ in the enumeration of the class induced by h is at step $m(i)$ (we refer to $m(i)$ as to the "least index" of $\varphi_{h(i)}$ in $\mathcal{L}_h$).

We are now in a position to introduce the two criteria of efficient identification informally described in the introduction. As specified before, they will be applied to uniform classes and they will be referred both to EX and NV identification paradigms.

Definition 5 Let $\mathcal{L}_h = \{\varphi_{h(i)} : i \in \omega\}$ be a uniform class. We say that:

(i) $\mathcal{L}_h$ is *EX-efficiently identifiable* ($\mathcal{L}_h \in EX^{\textit{eff}}$) if there exist $g \in P$ and a polynomial p such that, for every $\varphi_{h(i)} \in \mathcal{L}_h$, g EX-identifies $\varphi_{h(i)}$ in at most $p(m(i))$ guesses, i.e.:

$$(\exists n_0 < p(m(i)))(\forall n \geq n_0) g(<\varphi_{h(i)}(0), ..., \varphi_{h(i)}(n)>) = i'$$

for some $i' \in \omega$ such that $\varphi_{h(i)} = \varphi_{h(i')}$.

(ii) $\mathcal{L}_h$ is *NV-efficiently identifiable* ($\mathcal{L}_h \in NV^{\textit{eff}}$) if there exist $g \in P$ and a polynomial p such that, for every $\varphi_{h(i)} \in \mathcal{L}_h$, g NV-identifies $\varphi_{h(i)}$ in at most $p(m(i))$ guesses, i.e.:

$$(\exists n_0 < p(m(i)))(\forall n \geq n_0) g(<\varphi_{h(i)}(0), ..., \varphi_{h(i)}(n)>) = \varphi_{h(i)}(n+1).$$

Definition 6 Let $\mathcal{L}_h = \{\varphi_{h(i)} : i \in \omega\}$ be a uniform class. We say that:

(i) $\mathcal{L}_h$ is *EX-very efficiently identifiable* ($\mathcal{L}_h \in EX^{\textit{v-eff}}$) if there exist $g \in P$ and a polynomial p such that, for every $\varphi_{h(i)} \in \mathcal{L}_h$, g EX-identifies $\varphi_{h(i)}$ in at most $p(|m(i)|)$ guesses, i.e.:

$$(\exists n_0 < p(|m(i)|))(\forall n \geq n_0) g(<\varphi_{h(i)}(0), ..., \varphi_{h(i)}(n)>) = i'$$

for some $i' \in \omega$ such that $\varphi_{h(i)} = \varphi_{h(i')}$.

(ii) $\mathcal{L}_h$ is *NV-very efficiently identifiable* ($\mathcal{L}_h \in NV^{\textit{v-eff}}$) if there exist $g \in P$ and a polynomial p such that, for every $\varphi_{h(i)} \in \mathcal{L}_h$, g NV-identifies $\varphi_{h(i)}$ in at most $p(|m(i)|)$ guesses, i.e.:

$$(\exists n_0 < p(|m(i)|))(\forall n \geq n_0) g(<\varphi_{h(i)}(0), ..., \varphi_{h(i)}(n)>) = \varphi_{h(i)}(n+1).$$

The following relations between the previous paradigms are easily estabilished:

Proposition 2 *Let $\mathcal{L}_h = \{\varphi_{h(i)} : i \in \omega\}$ be a uniform class.*

(i) $\mathcal{L}_h \in EX^{\textit{eff}} \Rightarrow \mathcal{L}_h \in NV^{\textit{eff}}$.

(ii) $\mathcal{L}_h \in EX^{\textit{v-eff}} \Rightarrow \mathcal{L}_h \in NV^{\textit{v-eff}}$.

(iii) $\mathcal{L}_h \in EX^{\textit{v-eff}} \Rightarrow \mathcal{L}_h \in EX^{\textit{eff}}$.

(iv) $\mathcal{L}_h \in NV^{v\text{-}eff} \Rightarrow \mathcal{L}_h \in NV^{eff}$.

We can easily find uniform classes which are EX-very efficiently identifiable, hence EX-efficiently identifiable and NV-(very) efficiently identifiable. But the reverse of the latter two items of Proposition 2 does not hold, as shown in Example 1 (3) below.

Examples 1 (1) Let $\mathcal{C} = \{f_i : i \in \omega\}$ such that, for every $i, x \in \omega$, $f_i(x) = i$. If $\Psi(i,x) = i$, clearly $\Psi(i,x)$ is a universal function for $\mathcal{C}$. Moreover, if $h \in P$ is a Ψ-indexing for $\mathcal{C}$, $\varphi_{h(i)}(x) = f_i(x)$, $m(i) = i$, $\mathcal{C} = \mathcal{L}_h = \{\varphi_{h(i)} : i \in \omega\}$. Define, for every $a_0, ..., a_n \in \omega$, $g(< a_0, ..., a_n >) = a_0$. It is obvious that $g \in P$ and g EX-identifies (NV-identifies) every $\varphi_{h(i)} \in \mathcal{L}_h$ after the first guess.

(2) For every finite uniform class $\mathcal{L}_h$, $\mathcal{L}_h \in EX^{v\text{-}eff}$, suppose $card(\mathcal{L}_h) = k$ and let $\Psi(i,x)$ be a universal function for $\mathcal{L}_h$. Let $i_0, ..., i_{k-1}$ be such that, for every $r, r' \leq k-1$, if $r \neq r'$, then $\varphi_{h(i_r)} \neq \varphi_{h(i_{r'})}$ and $\mathcal{L}_h = \{\varphi_{h(i_0)}, ..., \varphi_{h(i_{k-1})}\}$. Clearly we can find n_0 such that, for every $r, r' \leq k-1$, $i_r, i_{r'} \leq n_0$ and there exists $x \leq n_0$ such that $\varphi_{h(i_r)}(x) \neq \varphi_{h(i_{r'})}(x)$. For every $a_0, ..., a_n \in \omega$ let:

$$g(< a_0, ..., a_n >) = \begin{cases} i & \text{where } i = \mu j \leq n[(\forall x \leq n)(\Psi(j,x) = a_x)], \\ & \quad \text{if such } i \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

It is easy to verify that $g \in P$ and that g identifies each function of $\mathcal{L}_h$ after n_0 examples.

(3) Let $\mathcal{C} = \{f_i : i \in \omega\}$ where $f_0(x) = 1$ and, for every $i \in \omega \setminus \{0\}$,

$$f_i(x) = \begin{cases} 1 & \text{if } x \leq i \\ 0 & \text{otherwise} \end{cases}$$

It is easy to prove that, for all i, f_i can be identified after i examples but, if $p(x)$ is any polynomial and $i > p(|i|)$, it cannot be identified with $p(|i|)$ examples.
In a similar way the reader can show that $\mathcal{L}_h \in NV^{eff}$ but $\mathcal{L}_h \notin NV^{v\text{-}eff}$.

Proposition 2 and Examples 1(2) suggest us to investigate EX-very efficient identifiability of infinite uniform classes. This is the aim of the next sections.

4 Uniform classes in $EX^{v\text{-}eff}$

Some remarkable infinite uniform classes turn out to be EX-very efficiently identifiable (hence NV-very efficiently identifiable). Two interesting examples are offered by the class of "remainder" functions and the class of polynomials with positive integer coefficients in one variable.

4.1 Class of remainder functions

Notation 2 For every $a, b, n \in \omega$, $n \neq 0, 1$, we denote by $re(a, n)$ the remainder of a divided by n. If a is congruent to b modulo n, we write $a \equiv_n b$, while, if n divides a, we write $n|a$. We denote by $M.C.D.(a, b)$ and $m.c.m.(a, b)$, respectively, the greatest common divisor and the least common multiple of a and b.

Notation 3 Let $R_{MOD} = \{f_x : x \in \omega\}$ where, for every $x, n \in \omega$,

$$f_x(n) = \begin{cases} 0 & \text{if } n=0 \text{ or } n=1 \\ re(x,n) & \text{otherwise} \end{cases}$$

Let:

$$\Psi(x,n) = \begin{cases} 0 & \text{if } n=0 \text{ or } n=1 \\ re(x,n) & \text{otherwise} \end{cases}$$

$\Psi(x, n)$ is a universal function for R_{MOD} and, if $h \in P$ is a Ψ-indexing, $f_x(n) = \varphi_{h(x)}(n)$ and $R_{MOD} = \mathcal{L}_h = \{\varphi_{h(x)} : x \in \omega\}$.

Proposition 3 *Let $\Psi(x, n) \in P$ be the universal function for R_{MOD} defined in Notation 3 and let h be a Ψ-indexing. Then $R_{MOD} = \mathcal{L}_h$ and:*

(i) $\mathcal{L}_h \in EX^{v\text{-}eff}$.

(ii) $\mathcal{L}_h \in NV^{v\text{-}eff}$.

The proof of Proposition 3 is based on some known results of Number Theory [Keng 82], which we briefly summarize below, and on Procedure 1.

Proposition 4 *There exists a p-time algorithm which computes the greatest common divisor of any two natural numbers (Euclid's Algorithm).*

Corollary 1 *The function $m.c.m.(x, y)$ is in P.*

Definition 7 Let $a, b \in \omega$. We call the *inverse of a modulo b* (if it exists), $z \in \omega$ such that $za \equiv_b 1$.

Proposition 5 *Let $a, b \in \omega$. If a and b are relatively prime, there exists the inverse of a modulo b. Such an element can be computed in time polynomial in $|a|$ and $|b|$.*

We now define a function g associating to every $< k_1, ..., k_n > \in Seq$ a natural number k'_n such that, if we let $a_n = m.c.m.(2, 3, ..., n+1)$ and if $< k_1, ..., k_n >$ is the code of the sequence of remainders of some $x \in \omega$ modulo $2, 3, ..., n+1$, then $k'_n < a_n$ and there exists x_n such that $x = a_n x_n + k'_n$.

PROCEDURE 1

Step 0 Let:

$$g(<\emptyset>) = 0 \qquad (k'_1 = 0).$$
$$g(<k_1>) = k_1 \qquad (k'_1 = k_1).$$

Step n+1 Suppose we have computed $g(<k_1, ..., k_n>)$ (therefore k'_n). We distinguish the following cases:

(i) There exists $i \leq n+1$ such that $k_i \geq i+1$. Let:

$$g(<k_1, ..., k_n, k_{n+1}>) = 0 \qquad (k'_{n+1} = 0)$$

(the input sequence is not the sequence of remainders of any number).

(ii) For every $i \leq n+1$, $k_i < i+1$.

(1) Compute $a_n = m.c.m.(2, 3, ..., n+1)$.
(2) Compute $c_n = M.C.D.(a_n, n+2)$.
(3) Compute: $e_n = \frac{n+2}{c_n}$, $\quad d_n = \frac{a_n}{c_n}$, $\quad f_n = \frac{k'_n - k_{n+1}}{c_n}$.
(4) Compute u_n such that $u_n d_n \equiv_{e_n} 1$.
(5) Compute s_n such that $0 \leq s_n < e_n$ and $s_n \equiv_{e_n} -u_n f_n$.
If $-u_n f_n \geq 0$, $s_n = re(-u_n f_n, e_n)$.
If $-u_n f_n < 0$, compute $re(u_n f_n, e_n) = r_n$.
If $r_n > 0$, let $s_n = e_n - r_n$.
If $r_n = 0$, let $s_n = 0$.

Let:

$$g(<k_1, ..., k_n, k_{n+1}>) = k'_{n+1} = re(a_n s_n + k'_n, a_n e_n).$$

It is easy to verify the existence of f_n and u_n as required by Procedure 1 $(ii)(3)$, (4). Moreover, regarding Procedure 1 $(ii)(5)$, note that in each case s_n satisfies the required property. In particular, when $-u_n f_n < 0$: if $r_n > 0$, then $s_n \equiv_{e_n} e_n - r_n \equiv_{e_n} -r_n \equiv_{e_n} -u_n f_n$, while if $r_n = 0$, then $s_n \equiv_{e_n} 0 \equiv_{e_n} u_n f_n \equiv_{e_n} -u_n f_n$. It is also straightforward to prove that $g \in P$, since it is defined by limited iteration involving p-time functions (as $M.C.D.(x, y)$ and $m.c.m.(x, y)$).

Proposition 6 *Let $x \in \omega$ and let $k_1, ..., k_n, k_{n+1}, ...$ be, respectively, the remainders of x modulo $2, ..., n+1, n+2....$ Let, for every $n \in \omega$, k'_n be the output of g on input $<k_1, ..., k_n>$ (g defined in Procedure 1). Then*

$$(\forall n)[(k'_n < a_n) \wedge (\exists x_n)(x = a_n x_n + k'_n)].$$

Proof By induction on n.
<u>$n=1$</u> Clearly $a_1=2$ and $k'_1=k_1=re(x,2)$. So there exists x_1 such that:

$$x=2x_1+k_1=a_1x_1+k'_1.$$

<u>Inductive step</u> Suppose that $k'_n<a_n$ and that there exists x_n such that:

$$x=a_nx_n+k'_n. \tag{1}$$

Since $x\equiv_{n+2}k_{n+1}$ $(k_{n+1}<n+2)$, there exists z such that:

$$x=(n+2)z+k_{n+1}. \tag{2}$$

By (1) and (2) it follows that:

$$x=a_nx_n+k'_n=(n+2)z+k_{n+1}. \tag{3}$$

If $c_n=M.C.D.(a_n,n+2)$, then $c_n|(n+2)z-a_nx_n$, hence $c_n|k'_n-k_{n+1}$. Define:

$$d_n=\frac{a_n}{c_n} \qquad e_n=\frac{n+2}{c_n} \qquad f_n=\frac{k'_n-k_{n+1}}{c_n}.$$

Clearly $a_{n+1}=a_ne_n$ and, as noted before, there exists the inverse of d_n modulo e_n, u_n (i.e. $u_nd_n\equiv_{e_n}1$). Let s_n be such that $0\leq s_n<e_n$ and $s_n\equiv_{e_n}-u_nf_n$. By definition of d_n, e_n, f_n and (3), we obtain $d_nc_nx_n+k'_n=e_nc_nz+k_{n+1}$, hence

$$e_nc_nz-d_nc_nx_n=k'_n-k_{n+1}=c_nf_n \Leftrightarrow e_nz-d_nx_n=f_n \Leftrightarrow$$

$$u_ne_nz-u_nd_nx_n=u_nf_n. \tag{4}$$

But $u_ne_nz\equiv_{e_n}0$, $u_nd_n\equiv_{e_n}1$ and $u_nf_n\equiv_{e_n}-s_n$, so, by (4), $x_n\equiv_{e_n}s_n$. Then there exists x'_n such that $x_n=e_nx'_n+s_n$ and, by (1),

$$x=a_ne_nx'_n+a_ns_n+k'_n. \tag{5}$$

Moreover, by definition of k'_{n+1}, there exists x''_n such that:

$$a_ns_n+k'_n=a_ne_nx''_n+k'_{n+1}. \tag{6}$$

Finally let $x_{n+1}=x'_n+x''_n$. By (5) and (6) we obtain:

$$x=a_ne_n(x'_n+x''_n)+k'_{n+1}=a_{n+1}x_{n+1}+k'_{n+1}.$$

q.e.d.

Proposition 3 *Let $\Psi(x,n)\in P$ be the universal function for R_{MOD} defined in Notation 3 and let h be a Ψ-indexing. Then $R_{MOD}=\mathcal{L}_h=\{\varphi_{h(x)}:x\in\omega\}$ and:*

(i) $\mathcal{L}_h\in EX^{v\text{-}eff}$.

(ii) $\mathcal{L}_h\in NV^{v\text{-}eff}$.

Proof (i) Let g be the function defined in Procedure 1. For every $b_0,...,b_n\in\omega$, let:

$$g'(<b_0,...,b_n>)=\begin{cases} 0 & \text{if } n=0 \text{ or } n=1 \\ g(<b_2,...,b_n>) & \text{otherwise}\end{cases}$$

It is obvious that $g'\in P$ $(g\in P)$. Let $\varphi_{h(x)}\in\mathcal{L}_h$ and let $k_1,k_2,...$ be the remainders of x modulo $2,3,...$ respectively. The infinite sequence $\{a_n\}_{n\in\omega}$ $(a_n=m.c.m.(2,...,n+1))$ is increasing, so there exists $n\in\omega$ such that $a_n>x$: let $n_0=min\{n\in\omega:a_n>x\}$. For every $n\geq n_0$, if $k'_n=g(<k_1,...,k_n>)$, there exists $x_n \in \omega$ such that $x = a_n x_n + k'_n$ (Proposition 6). Since $a_n > x$, it must be the case that $x_n = 0$ and $k'_n = x$. Hence, for every $n > n_0$,

$$g'(<\varphi_{h(x)}(0), ..., \varphi_{h(x)}(n)>) = g(<k_1, ..., k_{n-1}>) = k'_{n-1} = x,$$

and g EX-identifies $\varphi_{h(x)}$ in at most $n_0 + 2$ guesses.

We claim that $n_0 \leq 4|x|^2 - 1$. Let Π be the product of maximal powers of primes less or equal to $4|x|^2$ (i.e. for every prime number p such that $p \leq 4|x|^2$, we consider the greatest n such that $p^n \leq 4|x|^2$). Obviously $\Pi = m.c.m.(2, ..., 4|x|^2) = a_{4|x|^2-1}$. We want to show $\Pi > x$, proving our claim and Proposition 3 as well. Let $p \leq 2|x|$, p a prime number. If p^n is a maximal power of p such that $p^n \leq 4|x|^2$, $p^{n+1} > 4|x|^2$, so $p^n > 2|x|$. By the Prime Number Theorem [Keng 82], for every $\varepsilon > 0$ and for sufficiently large x, the number of prime numbers less or equal to $2|x|$, $\pi(2|x|)$, is such that:

$$\pi(2|x|) \geq \frac{2|x|}{log_2(2|x|)} \cdot (1-\varepsilon) \geq \frac{2|x|}{\frac{3}{2} \cdot log_2(|x|)} \cdot (1-\varepsilon)$$

(for $|x| > 3$). Letting for example $\varepsilon = \frac{1}{4}$, for sufficiently large x, we obtain $\pi(2|x|) \geq \frac{|x|}{log_2(|x|)}$, hence

$$\Pi \geq (2|x|)^{\frac{|x|}{log_2(|x|)}}. \qquad (1)$$

Since $\frac{1+log_2(|x|)}{log_2(|x|)} > 1$, $\frac{|x|}{log_2(|x|)} \cdot (1 + log_2(|x|)) > log_2 x$. But this is equivalent to saying $(2|x|)^{\frac{|x|}{log_2(|x|)}} > x$. Hence, by (1),

$$\Pi = a_{4|x|^2-1} \geq (2|x|)^{\frac{|x|}{log_2(|x|)}} > x.$$

It follows that $n_0 \leq 4|x|^2 - 1$, so g' EX-identifies $\varphi_{h(x)}$ in at most $4|x|^2+1$ guesses (more precisely in at most $4|x|^2 + x_0$ guesses for some constant x_0). Then $\mathcal{L}_h \in EX^{v\text{-}eff}$.

(ii) Immediate from (i) (Proposition 2(ii)).

q.e.d.

4.2 Class of polynomials

Let $\mathcal{P}^1$ be the class of polynomials with positive integer coefficients in one variable. $\mathcal{P}^1$ is a uniform class with universal function $\Psi(k, x) = \sum_{i=0}^{lth(k)-1} \beta(i+1, k) x^i$. For every $p \in \mathcal{P}^1$, if $p(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$, clearly $p(0) = a_0$, $p(1) = a_n + a_{n-1} + ... + a_0$ and $n \leq \lfloor log_2(p(2)) \rfloor$. The next procedure computes the degree of p and its coefficients in efficient time.

PROCEDURE 2

Let $\sigma \in Seq$ be given, say $\sigma = <b_0, ..., b_n>$. We start from the following observation: if there exists a polynomial $p(x)$ of degree $\leq n$ such that $b_0 = p(0)$, $b_1 = p(1)$,..., $b_n = p(n)$, there must be $u_0, ..., u_n \in \omega$ such that:

$$\begin{aligned}
&u_0 = b_0\\
&u_0 + u_1 + \ldots + u_n = b_1\\
&u_0 + 2u_1 + \ldots + 2^n u_n = b_2\\
&\vdots\\
&u_0 + nu_1 + \ldots + n^n u_n = b_n
\end{aligned} \tag{1}$$

Our learning algorithm is given by the function Φ which associates to each $\sigma = < b_0, \ldots, b_n >$ the code of the polynomial whose coefficients are given by the solution of the system (1) (if $u_k \neq 0$ and $u_{k+1} = u_{k+2} = \ldots = u_n = 0$ we output the code of the polynomial $u_0 + u_1 x + \ldots + u_k x^k$, i.e. we ignore the zero's). Note that the matrix of coefficients of system (1) described above has a Vandermonde's determinant, which is always non-zero. So the system (1) has a unique solution. Moreover, it is sufficient to find an algorithm that solves the system (1) in time polynomial in $|\sigma|$. Since $|\sigma| \geq |b_0| + |b_1| + \ldots + |b_n|$, such an algorithm is easy to give. Once $lth(\sigma)$ equals the degree of the polynomial to be identified plus 1, we reach the correct guess. Thus the number of examples we need to identify a polynomial is less than the length of the code of the polynomial itself.

5 A characterization of efficient identification

We want to characterize efficient identification of uniform classes in both identification paradigms. If $\mathcal{L}_h$ is a uniform class we indicate by $(*)$ the following condition:

$$(\exists \text{ polynomial } p)(\forall i)(\forall j < m(i))(\exists x \leq p(m(i)))(\varphi_{h(j)}(x) \neq \varphi_{h(i)}(x)).$$

Intuitively: in time polynomial in the "least index" for $\varphi_{h(i)}$ in $\mathcal{L}_h$, $m(i)$, we can distinguish $\varphi_{h(i)}$ from every $\varphi_{h(j)}$ with index less than $m(i)$.

Theorem 2 *Let $\mathcal{L}_h = \{\varphi_{h(i)} : i \in \omega\}$ be a uniform class.*

(i) $\mathcal{L}_h \in EX^{eff} \Leftrightarrow \mathcal{L}_h$ *satisfies* $(*)$.

(ii) $\mathcal{L}_h \in NV^{eff} \Leftrightarrow \mathcal{L}_h$ *satisfies* $(*)$.

Proof We prove (i). Proof of (ii) is similar, therefore it is left to the reader. Let $\Psi(i,x)$ be a universal function for $\mathcal{L}_h$.
$(\Leftarrow)$ Assume that $\mathcal{L}_h$ satisfies $(*)$. Let $p(x)$ be a polynomial such that

$$(\forall i)(\forall j < m(i))(\exists x \leq p(m(i)))(\varphi_{h(j)}(x) \neq \varphi_{h(i)}(x)).$$

Let g be such that, for every $a_0, \ldots, a_n \in \omega$:

$$g(< a_0, \ldots, a_n >) = \begin{cases} i & \text{where } i = \mu j \leq n[(\forall x \leq n)(\Psi(j,x) = a_x)], \\ & \quad \text{if such } i \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

$g \in P$. For every $< a_0, ..., a_n >$, g takes at most $(n+1)^2$ steps of computation of Ψ on input (j,x) with $j,x \leq n$ and each time it compares $\Psi(j,x)$ with a_x. Now $n < |< a_0, ..., a_n >|$ and $\Psi(i,x) \in P$, so the running time of g is bounded by a polynomial in $|< a_0, ..., a_n >|$. Let $\varphi_{h(i)} \in \mathcal{L}_h$. For every $n \geq p(m(i))$, $m(i) \leq n$ and if $j < m(i)$ there exists $x \leq n$ such that $\Psi(j,x) = \varphi_{h(j)}(x) \neq \varphi_{h(i)}(x)$ $((*))$. For every $x \in \omega$, $\Psi(m(i),x) = \varphi_{h(i)}(x)$. So for every $n \geq p(m(i))$,

$$g(< \varphi_{h(i)}(0), ..., \varphi_{h(i)}(n) >) = m(i).$$

Hence g EX-identifies $\varphi_{h(i)}$ in at most $p(m(i))+1$ guesses, and $\mathcal{L}_h \in EX^{eff}$.

($\Rightarrow$) Let $g \in P$ and the polynomial p be such that, for every $\varphi_{h(i)} \in \mathcal{L}_h$, g EX-identifies $\varphi_{h(i)}$ in at most $p(m(i))$ guesses. We show that $\mathcal{L}_h$ with p satisfies $(*)$. Assume by contradiction:

$$(\exists i)(\exists j < m(i))(\forall x \leq p(m(i)))(\varphi_{h(j)}(x) = \varphi_{h(i)}(x)). \tag{1}$$

If $j < m(i)$, then $\varphi_{h(j)} \neq \varphi_{h(i)}$. Let $\tilde{n} = min\{x \in \omega : \varphi_{h(j)}(x) \neq \varphi_{h(i)}(x)\}$. By (1), $\tilde{n} > p(m(i))$ and, obviously, $p(m(i)) \geq p(m(j))$. Since g EX-identifies $\varphi_{h(i)}$ in at most $p(m(i))$ guesses, $g(< \varphi_{h(i)}(0), ..., \varphi_{h(i)}(\tilde{n}-1) >) = i'$ for some $i' \in \omega$ such that $\varphi_{h(i)} = \varphi_{h(i')}$. By definition of $\tilde{n}$, for every $x \leq \tilde{n}-1$, $\varphi_{h(j)}(x) = \varphi_{h(i)}(x)$, hence $g(< \varphi_{h(j)}(0), ..., \varphi_{h(j)}(\tilde{n}-1) >) = i'$ where $\varphi_{h(i')} \neq \varphi_{h(j)}$. So g EX-identifies $\varphi_{h(j)}$ in a number of guesses $> \tilde{n} > p(m(j))$ contradicting the assumptions above. Therefore $\mathcal{L}_h$ with p satisfies $(*)$.

q.e.d.

Corollary 2 *Let $\mathcal{L}_h = \{\varphi_{h(i)} : i \in \omega\}$ be a uniform class. Then $EX^{eff} = NV^{eff}$.*

Proof Immediate by Theorem 2.

q.e.d.

Corollary 2 allows us to speak about efficient identification of a uniform class without specifying the particular identification paradigm we refer to. Moreover Theorem 2 provides a useful method for detecting classes that fail to be efficiently identifiable.

Example 1 Let $\mathcal{C} = \{f_i : i \in \omega\}$ where $f_0(x) = 0$ and, for every $i \in \omega \setminus \{0\}$,

$$f_i(x) = \begin{cases} 1 & \text{if } x = 2^i \\ 0 & \text{otherwise} \end{cases}$$

Define:

$$\Psi(i,x) = \begin{cases} 1 & \text{if } i \neq 0 \text{ and } x = 2^i \\ 0 & \text{otherwise} \end{cases}$$

$\Psi(i,x)$ is a universal function for $\mathcal{C}$ and, if h is a Ψ-indexing, for every $i,x \in \omega$, $\varphi_{h(i)}(x) = f_i(x)$, $m(i) = i$, $\mathcal{C} = \mathcal{L}_h = \{\varphi_{h(i)} : i \in \omega\}$.

It is immediate to verify that $\mathcal{L}_h$ does not satisfy condition $(*)$. Hence $\mathcal{L}_h$ is not efficiently identifiable.

An important application of Theorem 2 deals with efficient identification of the entire class P. We notice that P is not a uniform class, because there is no universal p-time function for P. But we can overcome this problem by considering the following:

Definition 8 Let $\mathcal{C} \subseteq P$. We say that $\mathcal{C}$ is *weakly uniform* if there exist $\Psi(i,x)$, $h(i)$ and a polynomial $p(i,x)$ such that:

(i) $\lambda ix.\Psi(i,x) \downarrow \leq p(i,x)$.

(ii) For every $i \in \omega$, $\lambda x.\Psi(i,x) \in \mathcal{C}$.

(iii) For every $f \in \mathcal{C}$, there exists $i \in \omega$ such that $f = \lambda x.\Psi(i,x)$.

(iv) h is strictly increasing and, for every $i \in \omega$, $\lambda x.\Psi(i,x) = \varphi_{h(i)}(x)$.

Such a function $\Psi(i,x)$ is called a *universal function for* $\mathcal{C}$.

Obviously P is a weakly uniform class by taking some acceptable indexing Φ for P as universal function and the identity function as Φ-indexing. Moreover Definition 5 and Definition 6 can be applied as well to weakly uniform classes and it is easy to verify that Theorem 2 remains true for these classes too. So P is efficiently identifiable if and only if P satisfies condition $(*)$. We use this result to prove the following:

Theorem 3 *P is not efficiently identifiable with respect to any acceptable indexing.*

The proof of Theorem 3 is based on the following propositions, in which we refer to any acceptable indexing $\Phi(i,x)$ for P and to the identity function as a Φ-indexing, so that $P = \{\lambda x.\Phi(i,x) : i \in \omega\} = \{\varphi_i : i \in \omega\}$.

Proposition 7 *If P is efficiently identifiable with respect to $\Phi(i,x)$ then P satisfies $(*)$, i.e.:*

$$(\exists \text{ polynomial } p)(\forall i)(\forall j < m(i))(\exists x \leq p(m(i)))(\varphi_j(x) \neq \varphi_i(x)).$$

Proof Immediate from Theorem 2, since P is weakly uniform.

q.e.d.

Proposition 8 *Let $M = \{m(i) : i \in \omega\}$ be the set of the least Φ-indexes for functions in P. If P satisfies $(*)$, then M is recursive.*

Proof It suffices to notice that

$$M = \{m(i) : i \in \omega\} = \{i \in \omega : (\forall j < i)(\exists x \leq p(i))(\Phi(j,x) \neq \Phi(i,x))\}$$

is an EXP-TIME set, since it is defined by EXP-TIME relations with polynomially bounded quantifiers. So M is recursive.

q.e.d.

Proposition 9 *Let $M=\{m(i):i\in\omega\}$ be the set of the least Φ-indexes for functions in P. If M is recursive then $R'=\{(i,j):\varphi_i=\varphi_j\}$ is a recursive relation.*

Proof First of all, notice that $R'=\{(i,j):m(i)=m(j)\}$, so it suffices to show that $m(i)$ is a recursive function. For every $i\in\omega$, let $M_i=\{j\in M:j\leq i\}$. M_i is a finite non empty set ($m(i)\in M_i$). Moreover, if $j,j'\in M_i$, $j\neq j'$, then $\varphi_j\neq\varphi_{j'}$, because M_i contains only least indexes of functions in P (less than or equal to i). $m(i)$ can be computed by eliminating from M_i all indexes j such that $\varphi_j\neq\varphi_i$. We proceed by steps: at each step n we compute all functions with an index in M_i till input n, eliminating those which differ from φ_i on some of these input. More precisely, if $M_i=\{j_0,...,j_k\}$, we let:

$M_i^0=M_i$, $\quad M_i^{n+1}=\{j\in M_i:(\forall x\leq n+1)(\varphi_j(x)=\varphi_i(x))\}$.

If $j\in M_i$ and $\varphi_j\neq\varphi_i$ there exists $n\in\omega$ such that $\varphi_j(n)\neq\varphi_i(n)$, so $j\notin M_i^n$. For sufficiently large n, M_i^n contains one element only: this will be the value assumed by $m(i)$.

q.e.d.

Corollary 3 *If P is efficiently identifiable with respect to $\Phi(i,x)$ then $R'=\{(i,j):\varphi_i=\varphi_j\}$ is a recursive relation.*

Proof Immediate by Proposition 7, Proposition 8, Proposition 9.

q.e.d.

We are now ready to prove Theorem 3.

Theorem 3 *P is not efficiently identifiable with respect to any acceptable indexing.*

Proof Let $K=\{x\in\omega:\pi_x(x)\downarrow\}$. By the MRDP-Theorem [Davis 58] there exists $R(x,y)\in P$ such that $K=\{n\in\omega:\exists yR(n,y)\}$. Let:

$$\Psi(n,x)=\begin{cases}0 & \text{if } (\forall y\leq|x|)\neg R(n,y)\\ 1 & \text{otherwise}\end{cases}$$

Obviously $\Psi(n,x)\in P$. Assume for a contradiction that P is efficiently identifiable with respect to some acceptable indexing $\Phi(i,x)$. Then $R'=\{(i,j):\varphi_i=\varphi_j\}$ is a recursive relation (Corollary 3). Moreover, since $\Psi(n,x)\in P$, there exists $h\in P$ such that, for every $n,x\in\omega$, $\Psi(n,x)=\varphi_{h(n)}(x)$ and:

$$\lambda x.\Psi(n,x)=0\Leftrightarrow(\forall x)(\forall y\leq|x|)\neg R(n,y)\Leftrightarrow(\forall y)\neg R(n,y)\Leftrightarrow n\notin K. \quad (1)$$

Let $n_0\notin K$ (for instance, let n_0 be an index for the empty function). By (1), $\lambda x.\Psi(n_0,x)=0$ and so, for every $i\in\omega$:

$$R'(h(i),h(n_0))\Leftrightarrow\varphi_{h(i)}=\varphi_{h(n_0)}\Leftrightarrow\lambda x.\Psi(i,x)=\lambda x.\Psi(n_0,x)=0. \quad (2)$$

By (1) and (2) it follows:

$$i \notin K \Leftrightarrow \lambda x.\Psi(i,x)=0 \Leftrightarrow R'(h(i),h(n_0)),$$

and $\overline{K}$ is a recursive set, a contradiction. So P is not efficiently identifiable with respect to any acceptable indexing.

q.e.d.

6 EX-very efficient identification

The problem of finding a characterization for EX-very efficient identification turns out to be more complex than in the case of efficient identification. We consider a condition, $(*')$, which is the natural reduction of condition $(*)$ to the very efficient setting. We shall see that, even though $(*')$ remains a necessary condition for EX-very efficient identification, its sufficiency is equivalent to $\mathcal{P}=\mathcal{NP}$. The reason for such a discrepancy with respect to the efficient case is to be found in the fact that, even if each function f in a given uniform class can be distinguished by any other of lesser index in time polynomial in the length of its least index, the learner may not output the least index of f in time polynomial in the length of the input (the least index of f can be in general too large with respect to this length). This is the content of Theorem 4 and Theorem 5.

Let $\mathcal{L}_h$ be a uniform class. Let $(*')$ denote the following condition:

$$(\exists \text{ polynomial } p)(\forall i)(\forall j<m(i))(\exists x \le p(|m(i)|))(\varphi_{h(j)}(x) \neq \varphi_{h(i)}(x)).$$

Theorem 4 *Let $\mathcal{L}_h=\{\varphi_{h(i)} : i\in\omega\}$ be a uniform class. Then:*

(i) $\mathcal{L}_h \in EX^{v\text{-}eff} \Rightarrow \mathcal{L}_h$ *satisfies* $(*')$.

(ii) $\mathcal{L}_h \in NV^{v\text{-}eff} \Rightarrow \mathcal{L}_h$ *satisfies* $(*')$.

Proof Analogous to Theorem 2.

q.e.d.

Theorem 4 provides a useful method to prove that uniform classes are not very efficiently identifiable.

Example 2 Let $\mathcal{L}_h=\{\varphi_{h(i)} : i\in\omega\}$ be the class defined in Examples 1(3). We proved that it is not very efficiently identifiable in any identification paradigm. The same result follows by observing that $\mathcal{L}_h$ does not satisfy $(*')$.

We now show that the reverse implication of Theorem 4 is a “hard” problem.

Theorem 5 *The following are equivalent:*

(i) Every uniform class satisfying $(*')$ is EX-very efficiently identifiable.

(ii) $\mathcal{P}=\mathcal{NP}$.

Proof $(i) \Rightarrow (ii)$ Consider the problem of deciding whether or not a propositional formula A in conjunctive normal form ($A \in CNF$) is satisfiable ($A \in SAT$). This problem is $\mathcal{NP}$-complete. We want to show that if $(*')$ is sufficient for EX-very efficient identification of uniform classes, then SAT is solvable with a deterministic polynomial time algorithm. This will imply $\mathcal{P} = \mathcal{NP}$. For every $A \in CNF$, let $Var_A = \{x_i : \text{either } x_i \text{ or } \overline{x_i} \text{ occurs in } A\}$. We call $\tau : Var_A \to \{0,1\}$ a truth assignment on A and we let $Ass_A = \{\tau : \tau \text{ truth assignment on } A\}$. Moreover we denote by:

- $[\lceil A \rceil]_i$ $(i \leq |\lceil A \rceil|)$, the i-th bit in the binary expansion of $\lceil A \rceil$.
- $\tau(A)$, the truth value of A on assignment τ; $|\tau|$, the cardinality of $rng(\tau)$.
- $\lceil A \rceil$ $(\lceil \tau \rceil)$, the code of A (τ).

If $\tau, \tau' \in Ass_A$, then $|\tau| = |\tau'|$. Moreover, a truth assignment can be regarded as a finite 0–1 sequence, hence it is coded as described in Section 2.1. We consider formulas in conjunctive normal form in the alphabet $\Sigma = \{x, |, \wedge, \vee, \_, (,)\}$ and we suppose them coded as shown in Preliminaries. We recall that, for every $n \in \omega$, we can check if n is the code of some $A \in CNF$ and then decode it in time polynomial in $|n|$. The same properties hold for $n \in COD$, where $COD = \{ < \lceil A \rceil, \lceil \tau \rceil >: A \in CNF, \tau \in Ass_A\}$. We finally note that, if $n = < \lceil A \rceil, \lceil \tau \rceil >$, $|n| = 2(|\lceil A \rceil| + |\lceil \tau \rceil| + 1)$, so $|\lceil A \rceil| \leq |n| \leq 4|\lceil A \rceil| + 2$. Moreover, for some polynomial p', $|\lceil A \rceil| \leq p'(lth(A))$.

Let $\Psi(n, x)$ be such that, for every $n, x \in \omega$, the following conditions hold:

- if $n \in COD$, say $n = < \lceil A \rceil, \lceil \tau \rceil >$ for some $A \in CNF$, $\tau \in Ass_A$, let

$$\Psi(n, x) = \begin{cases} [\lceil A \rceil]_x & \text{if } x < |\lceil A \rceil| \\ 1 & \text{if } x = |\lceil A \rceil| + 4 \text{ and } \tau(A) = 1 \\ 0 & \text{otherwise} \end{cases}$$

- if $n \notin COD$, let $\Psi(n, x) = 0$.

Intuitively, if $n \in COD$ and $n = < \lceil A \rceil, \lceil \tau \rceil >$, $\Psi(n, x)$ assumes values:

$$[\lceil A \rceil]_0, ..., [\lceil A \rceil]_{|\lceil A \rceil|-1}, 0, 0, 0, 0, \tau(A), 0, 0.....$$

where $\tau(A) = 0$ or $\tau(A) = 1$. By the previous remarks, it follows $\Psi(n, x) \in P$. If $h \in P$ is a Ψ-indexing, then for every $n, x \in \omega$, $\Psi(n, x) = \varphi_{h(n)}(x)$, and $\mathcal{L}_h = \{\varphi_{h(n)} : n \in \omega\}$ is uniform. We verify that $\mathcal{L}_h$ satisfies $(*')$. Let $\varphi_{h(n)} \in \mathcal{L}_h$. If $n \notin COD$, then $m(n) = 0$ and $(*')$ is trivial. If $n \in COD$, say $n = < \lceil A \rceil, \lceil \tau \rceil >$ for some $A \in CNF$, $\tau \in Ass_A$, consider $j < m(n)$:

- If $j \notin COD$, then $\Psi(j, 0) = 0$, while $\Psi(n, 0) = [\lceil A \rceil]_0 = 1$. So $(*')$ is satisfied.
- If $j \in COD$, say $j = < \lceil A' \rceil, \lceil \tau' \rceil >$ for some $A' \in CNF$, $\tau' \in Ass_{A'}$, then we can distinguish the following cases:

(1) $|\lceil A' \rceil| < |\lceil A \rceil|$: there exists $x < |\lceil A \rceil| + 1$ such that $\Psi(j, x) \neq \Psi(n, x)$.

(2) $|\lceil A \rceil| < |\lceil A' \rceil|$: there exists $x < |\lceil A' \rceil| + 1$ such that $\Psi(j, x) \neq \Psi(n, x)$.

(3) $|\lceil A'\rceil| = |\lceil A\rceil|$: if $\lceil A'\rceil \neq \lceil A\rceil$, then there is $x \leq |\lceil A\rceil|$ such that $\Psi(j,x) \neq \Psi(n,x)$. If $\lceil A'\rceil = \lceil A\rceil$, then $\tau' \in Ass_A$, $|\tau| = |\tau'|$, $\tau'(A) \neq \tau(A)$, so $\Psi(j, |\lceil A\rceil| + 4) \neq \Psi(n, |\lceil A\rceil| + 4)$.

In cases (1), (3) it results that $|\lceil A\rceil| + 1 \leq |m(n)|$, while in case (2), since $j < m(n)$, $|\lceil A'\rceil| + 1 \leq |< \lceil A'\rceil, \lceil \tau'\rceil >| = |j| \leq |m(n)|$. So $\mathcal{L}_h$ satisfies $(*')$ with $p(x) = x$.

By hypothesis, $(*')$ is sufficient for EX-very efficient identification of uniform classes, so $\mathcal{L}_h \in EX^{v\text{-}eff}$. Let $g \in P$ and p be such that, for every $\varphi_{h(n)} \in \mathcal{L}_h$, g EX-identifies $\varphi_{h(n)}$ in at most $p(|m(n)|)$ guesses. For every $A \in CNF$, consider $\sigma^A \in Seq$ such that $lth(\sigma^A) = p(4|\lceil A\rceil| + 2)$ and

$$\sigma^A = < [\lceil A\rceil]_0, ..., [\lceil A\rceil]_{|\lceil A\rceil|-1}, 0, 0, 0, 0, 1, 0, ..., 0 > .$$

Let $g(\sigma^A) = m$. If $A \in SAT$, then $m = < \lceil A\rceil, \lceil \tau\rceil >$ for some $\tau \in Ass_A$ such that $\tau(A) = 1$. In fact, by assumption, g EX-identifies $\varphi_{h(<\lceil A\rceil, \lceil \tau\rceil>)}$ in at most $p(|m(< \lceil A\rceil, \lceil \tau\rceil >)|)$ guesses, and $p(|m(< \lceil A\rceil, \lceil \tau\rceil >)|) \leq p(4|\lceil A\rceil| + 2) = lth(\sigma^A)$. On the other hand, if $m = < \lceil A\rceil, \lceil \tau\rceil >$ and $\tau(A) = 1$, then $A \in SAT$. So:

$$A \in SAT \iff g(\sigma^A) = < \lceil A\rceil, \lceil \tau\rceil > \quad \text{with } \tau \in Ass_A, \tau(A) = 1. \tag{1}$$

It is straightforward to verify that, given $A \in CNF$ and the polynomial p, $g(\sigma^A)$ and $p(4|\lceil A\rceil| + 2)$ can be computed in time polynomial in $|\lceil A\rceil|$. Now, if $g(\sigma^A) = m$, we can check if $m \in COD$ and if $(m)_1 = \lceil A\rceil$ in time polynomial in $|m|$ ($|m| \leq |\lceil A\rceil|$). Moreover, if $(m)_2 = \tau$ ($\tau \in Ass_A$), we can decode $\lceil A\rceil$ and compute $\tau(A)$ in time polynomial in $|\lceil A\rceil|$. But $|\lceil A\rceil| \leq p'(lth(A))$ for some polynomial p'. So, by (1), the problem of deciding if $A \in SAT$ is solvable in time polynomial in $lth(A)$.

$(ii) \Rightarrow (i)$ Let $\mathcal{L}_h = \{\varphi_{h(i)} : i \in \omega\}$ be a uniform class with universal function $\Psi(i,x)$. Suppose that $\mathcal{L}_h$ satisfies $(*')$ by p, i.e.:

$$(\forall i)(\forall j < m(i))(\exists x \leq p(|m(i)|))(\varphi_{h(j)}(x) \neq \varphi_{h(i)}(x)).$$

Let g be such that, for every $a_0, ..., a_n \in \omega$:

$$g(< a_0, ..., a_n >) = \begin{cases} i & \text{where } i = \mu j \leq 2^n[(\forall x \leq n)(\Psi(j,x) = a_x)], \\ & \quad \text{if such } i \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

g can be computed by binary search in polynomial-time using the NP oracle:

$$R(\sigma, y, z) \equiv (y + z \leq \sigma) \wedge (\exists i \leq y + z) \\ [(y \leq i) \wedge (\forall x < lth(\sigma))(\sigma_x = \Psi(i,x))].$$

If $\mathcal{P} = \mathcal{NP}$, then R is in P, hence $g \in P$. Let $\varphi_{h(i)} \in \mathcal{L}_h$. If $j < m(i)$, by $(*')$, there exists $x \leq p(|m(i)|)$ such that $\varphi_{h(j)}(x) \neq \varphi_{h(i)}(x)$. Moreover $m(i) < 2^{p(|m(i)|)}$. So, for every $n \geq p(|m(i)|)$,

$$g(< \varphi_{h(i)}(0), ..., \varphi_{h(i)}(n) >) = m(i)$$

and g EX-identifies $\varphi_{h(i)}$ in at most $p(|m(i)|) + 1$ guesses. This concludes the proof of the theorem.

q.e.d.

We conclude this section with a result concerning the relations between $NV^{v\text{-}eff}$ and $EX^{v\text{-}eff}$. First of all, We have thus shown:

Theorem 6 *The following are equivalent:*

(i) $EX^{v\text{-}eff} = NV^{v\text{-}eff}$.

(ii) $\mathcal{P} = \mathcal{NP}$.

Proof $(i) \Rightarrow (ii)$ Suppose $\mathcal{P} \neq \mathcal{NP}$. The class $\mathcal{L}_h$ defined in the proof of Theorem 5 is clearly in $NV^{v\text{-}eff}$. In fact the learner Φ, defined by $\Phi(\sigma) = 0$, NV identifies each $\varphi_{h(i)} \in \mathcal{L}_h$ upon seeing $\leq |i|$ examples. Moreover we have seen that $\mathcal{L}_h$ satisfies $(*')$, hence, by the initial assumption, $\mathcal{L}_h \notin EX^{v\text{-}eff}$ (Theorem 5). It follows that $EX^{v\text{-}eff} \neq NV^{v\text{-}eff}$.

$(ii) \Rightarrow (i)$ If $\mathcal{P} = \mathcal{NP}$, then condition $(*')$ characterizes both $EX^{v\text{-}eff}$ and $NV^{v\text{-}eff}$ (Theorem 4, Theorem 5, Proposition 2(ii)). Hence $EX^{v\text{-}eff} = NV^{v\text{-}eff}$.

q.e.d.

References

[Aho, Hopcroft and Ullman 74] Aho, A.V., Hopcroft, J.E. and Ullman, J.D.: "The Design and Analysis of Computer Algorithms"; Addison-Wesley Publishing Company (1974).

[Angluin 87] Angluin, D.: "Learning Regular Sets from Queries and Counter-examples"; Information and Computation, 75 (1987), 87-106.

[Angluin 78] Angluin, D.: "On the Complexity of Minimum Inference of Regular Sets"; Information and Control, 39 (1978), 337-350.

[Angluin 81] Angluin, D.: "A Note on the Number of Queries Needed to Identify Regular Languages"; Information and Control, 51 (1981), 76-87.

[Angluin 80] Angluin, D.: "Inductive Inference of Formal Languages from Positive Data"; Information and Control, 45 (1980), 117-135.

[Angluin 88] Angluin, D.: "Identifying Languages from Stochastic Examples"; Information and Control (1988).

[Buss 86] Buss, S.R.: "Bounded Arithmetic"; Studies in Proof Theory Lecture Notes (1986).

[Gold 67] Gold, E.M.: "Language Identification in the Limit"; Information and Control, 10 (1967), 447-474.

[Keng 82] Keng, H.L.: "Introduction to Number Theory"; Springer-Verlag, Berlin, Heidelberg, New York (1982).

[Kearns and Vazirani 94] Kearns, M.J., Vazirani, U.V.: "An Introduction to Computational Learning Theory"; The MIT Press, Cambridge, Massachusetts, London, England (1994).

[Montagna 98] Montagna, F.: "Investigations in Measure One Learning of Classes of Languages"; Information and Computation, (1998).

[Montagna and Simi 97] Montagna, F., Simi, G.: "Paradigms in Measure Theorethic Learning and in Informant Learning"; to appear in Studia Logica (1997).

[Odifreddi 99] Odifreddi, P.: "Classical Recursion Theory II"; Elsevier, Amsterdam, 1999.

[Osherson, Stob and Weinstein 86] Osherson, D.N., Stob, M. and Weinstein, S.: "Systems that Learn. An Introduction to Learning Theory for Cognitive and Computer Scientists"; MIT Press, Cambridge MA (1986).

[Oxtoby 71] Oxtoby, J.C.: "Measure and Category. Graduate Texts in Mathematics"; New York, Springer-Verlag (1971).

[Pitt and Valiant 88] Pitt, L., Valiant, L.G.: "Computational Limitations on Learning from Examples"; J. ACM, 35, 4 (1988), 965-984.

[Pitt 89] Pitt, L.: "Probabilistic Inductive Inference"; J. ACM, 36, 4 (1989), 383-433.

[Rogers 67] Rogers, H.: "Theory of Recursive Functions and Effective Computability"; McGraw-Hill Book Company, New York, St. Louis, San Francisco, Toronto, London, Sydney (1967).

[Valiant 84] Valiant, L.G.: "A Theory of the Learnable"; Commun. ACM 27, 11, 4 (1984), 1134-1142.