INFORMATION AND COMPUTATION **139**, 234–257 (1997)
ARTICLE NO. IC972666

Time Bounded Frequency Computations

Maren Hinrichs* and Gerd Wechsung

Institut für Informatik, Friedrich-Schiller-Universität Jena, 07743 Jena, Germany
E-mail: meh@rz.uni-jena.de, wechsung@minet.uni-jena.de

(1) We obtain two new results concerning the inclusion problem of polynomial time frequency classes with equal numbers of errors.

1. $(m, m+d)\,\mathrm{P} \supsetneq (m+1, m+d+1)\,\mathrm{P}$ for $m < 2^d$.

2. $(m, m+d)\,\mathrm{P} = (m+1, m+d+1)\,\mathrm{P}$ for $m \geqslant c(d)$ where $c(d)$ is large enough.

This disproves a conjecture of Kinber. (2) We give a transparent proof of a generalization of Kinber's result that there exist arbitrarily complex problems admitting a polynomial time frequency computation. Several corollaries provide more insight into the structure of the hierarchy of polynomial time frequency classes. (3) The relationships between polynomial time frequency classes and selectivity classes are studied. © 1997 Academic Press

1. INTRODUCTION

The notion of frequency computation was introduced by Rose [Ros60] and McNaughton [McN61] in the early sixties and developed by Trakhtenbrot [Tra63], Kinber [Kin75] and Degtev [Deg81]. Let m and n be natural numbers with $n \geqslant m \geqslant 1$. A function $f\colon \Sigma^* \to \mathbb{N}$ is called (m, n)-computable if there exists a recursive operator $T\colon (\Sigma^*)^n \to \mathbb{N}^n$ with the following property. If $T(x_1, ..., x_n) = (y_1, ..., y_n)$ for mutually distinct numbers x_i then at least m of the equalities $f(x_i) = y_i$, $i = 1, ..., n$ are true. Then we also say that f is (m, n)-computed by T or T provides an (m, n)-computation for f.

We study frequency computations under time restrictions; i.e., we demand that the operator T be time bounded on input $x_1, ..., x_n$ by some function $t(\max |x_i|)$.

Let us define $(m, n)\,\mathrm{TIME}(t)$ to be the class of all predicates which are (m, n)-computable in time t and (m, n) to be the class of all predicates which are (m, n)-computable without any time restriction. In particular, $(m, n)\,\mathrm{P}$ denotes the class of predicates which are (m, n)-computable within polynomial time.

One of the most fundamental questions is to determine for which pairs $\langle m, n\rangle$ and $\langle h, k\rangle$ of natural numbers ($m \leqslant n$, $h \leqslant k$) the corresponding classes (m, n) and

* Supported by a grant from the Konrad-Adenauer-Stiftung.

0890-5401/97 $25.00

(h, k) (or $(m, n)\,\mathrm{TIME}(t)$ and $(h, k)\,\mathrm{TIME}(t)$ for given recursive function t) are equal (equality problem) or are comparable with respect to set inclusion (inclusion problem). It turns out that the solution of these problems depends on whether the classes are time bounded or not. The first step towards solving these problems was made by Trakhtenbrot [Tra63] who proved that $(m, n) = \mathrm{REC}$ (= the class of all recursive predicates) if $m/n > 1/2$. For $m/n \leqslant 1/2$, the classes (m, n) contain non-recursive predicates. For these classes the equality problem has been attacked by Degtev [Deg81]. His approach has been successfully completed by Kummer and Stephan [KS95]:

$$\text{For} \quad m/n \leqslant 1/2, \qquad (m, n) = (h, k) \Leftrightarrow m = h \wedge n = k.$$

In the case of time bounded frequency classes a solution of the inclusion problem (and hence of the equality problem) has been given by Kummer and Stephan [KS95] and McNicholl [McNic95]. The corresponding result for the polynomial time case is formulated as Theorem 2.3. It gives an equivalent reformulation of the inclusion problem in terms of the combinatorial notion of admissibility. It is, however, not always easy to apply this criterion to decide a specific inclusion question.

This is why we once more deal with this topic. Our goal is to determine explicitly under what conditions two given classes $(m, n)\,\mathrm{P}$ and $(h, k)\,\mathrm{P}$ are comparable with respect to set inclusion. We cannot give a full solution, but for classes with equal numbers of errors (i.e., $n - m = k - h$) we prove two theorems by shedding some light on the inclusion structure of the polynomial time frequency classes. We obtain:

Theorem 1.1.

$$(m, m+d)\,\mathrm{P} \supsetneq (m+1, m+d+1)\,\mathrm{P} \qquad \textit{for} \quad m < 2^d.$$

So the conjecture of Kinber that $(m, n)\,\mathrm{P} = (h, k)\,\mathrm{P}$ for $m/n \geqslant 2/3$ and $h/k \geqslant 2/3$ and $n - m = k - h$ is disproved. We cannot say exactly when equality holds, but we prove

Theorem 1.2. *For all $d \in \mathbb{N}$ there exists a number $c(d)$ such that $(m, m+d)\,\mathrm{P} = (m+1, m+d+1)\,\mathrm{P}$ for all $m \geqslant c(d)$.*

We conjecture that one may even take $c(d) = 2^d$, but our proof yields a much weaker bound.

In Section 3 we consider whether one can save time by allowing errors. Kinber [Kin75] answered this question positively. We prove the following more general result.

Theorem 1.3.. *Assume $1 \leqslant m \leqslant n$, $1 \leqslant d, 0 \leqslant k$ for m, n, k, $d \in \mathbb{N}$. Let t be an arbitrary recursive function. Then*

$$(m+k, n+k+d)\,\mathrm{P} \setminus (m, n)\,\mathrm{TIME}(t) \neq \emptyset.$$

In Section 4 we deal with the relationship between frequency classes and selectivity classes which were introduced by Hemaspaandra *et al.* [HJRW95] in that general form. A motivation for studying this relationship has to do with the question of whether there exist NP-complete problems that are polynomial time frequency computable. For more details see Section 4.

2. THE HIERARCHY OF TIME BOUNDED FREQUENCY-CLASSES

We consider the inclusion and the equality problem for classes $(m, n)\,\mathrm{TIME}(t)$ and $(m', n')\,\mathrm{TIME}(t)$ for $m/n > 1/2$ and $m'/n' > 1/2$. As Trakhtenbrot showed, $(m, n) = \mathrm{REC}$ for $m/n > 1/2$ one would expect $(m, n)\,\mathrm{P} = \mathrm{P}$. This is, however, not true, as proved by Kinber [Kinb75]. In the next section we give a transparent proof of his result that there exist arbitrarily complex sets in $(n-1, n)\,\mathrm{P}$.

For ease of notation we state our results in this section for polynomial time bounds. It is easily seen that they can be generalized to arbitrary time bounds t. We start with some easy inclusions of $(m, n)\,\mathrm{P}$-classes.

THEOREM 2.1.

1. $(m, n)\,\mathrm{P} \subseteq (m, n')\,\mathrm{P}$ *for* $n \leqslant n'$
2. $(m, n)\,\mathrm{P} \subseteq (km, kn)\,\mathrm{P}$ *for* $k \geqslant 1$
3. $(m+k, n+k)\,\mathrm{P} \subseteq (m, n)\,\mathrm{P}$ *for* $k \geqslant 0$.

The following definition introduces a local version of (m, n)-computability. It was given by Degtev [Deg81] and leads to a very natural combinatorial view on the inclusion and the equality problem of frequency classes.

DEFINITION 2.2. Assume $1 \leqslant m \leqslant n \leqslant s$ for $m, n, s \in \mathbb{N}$. A set $V \subseteq \{0, 1\}^s$ of binary vectors of length s is called (m, n)-admissible iff for every n numbers $1 \leqslant x_1 < \cdots < x_n \leqslant s$ there exists a vector $(y_1, y_2, ..., y_n)$ such that for each $v \in V$ at least m of the n equations $v(x_i) = y_i$ are true.

Here $v(k)$ denotes the kth component of v. One could say that a set V is (m, n)-admissible if it locally admits a solution.

Let $V \subseteq \{0, 1\}^s$ be (m, n)-admissible. For each sequence $1 \leqslant x_1 < \cdots < x_n \leqslant s$ and corresponding vector $(y_1, y_2, ..., y_n)$ we define a vector $z \in \{0, 1, *\}^s$ by

$$z(i) = \begin{cases} y_i & \text{if} \quad i = x_i \\ * & \text{otherwise.} \end{cases}$$

These z are considered to be rows of a table T. For vectors $u, v \in \{0, 1, *\}^s$ we define their Hamming distance by

$$d(u, v) = |\{i : u(i) \neq v(i) \wedge u(i) \neq * \wedge v(i) \neq *\}|.$$

Evidently, V is (m, n)-admissible if and only if there exists a table T with the following properties:

1. For each sequence $1 \leqslant x_1 < \cdots < x_n \leqslant s$ there exists exactly one vector $z \in T$ with $z(i) = *$ if and only if $i \notin \{x_1, ..., x_n\}$.

2. $d(u, v) \leqslant n - m$ for all $u \in T$ and $v \in V$.

Thus we say that V is (m, n)-admissible via T.

Now we give a theorem of Kummer and Stephan [KS95] which reduces the inclusion problem for classes $(m, n)\mathrm{P}$ and $(m', n')\mathrm{P}$ to finite combinatorics.

THEOREM 2.3 [KS95]. *For* $1 \leqslant m \leqslant n$ *and* $1 \leqslant m' \leqslant n'$, $(m, n)\mathrm{P} \subseteq (m', n')\mathrm{P}$ *iff any* (m, n)*-admissible set* $V \subseteq \{0, 1\}^{\max(n, n')}$ *is* (m', n')*-admissible.*

We use this reformulation of the equality problem for classes $(m, n)\mathrm{P}$ to prove Theorems 1.1 and 1.2 via Propositions 2.4 and 2.6.

PROPOSITION 2.4. *Let* $m < 2^d$. *Then there exists a set* $V \subseteq \{0, 1\}^{m+d+1}$ *which is* $(m, m+d)$*-admissible but not* $(m+1, m+d+1)$*-admissible.*

Before we prove this proposition let us consider an example of the construction of a set which is $(m, m+d)$-admissible but not $(m+1, m+d+1)$-admissible in order to obtain the main ideas of such a construction. This may cause an easier understanding of the general proof. Let us consider an extreme case, for instance $d = 2$ and $m = 2^d - 1 = 3$.

First we choose a set $W \subseteq \{0, 1\}^d = \{0, 1\}^2$ with $|W| = m + 1 = 4$. In this case we have only the following possibility:

$$W = \{w_1, w_2, w_3, w_4\} = \{11, 10, 01, 00\}.$$

Now we construct a set $V \in \{0, 1\}^6$ which is $(3, 5)$-admissible but not $(4, 6)$-admissible. Let

$$V = V_1 \cup V_2$$

with

$$V_1 = \{v_1, v_2, v_3, v_4\} = \{11\,0000, 10\,0000, 01\,0000, 00\,0000\}$$

$$V_2 = \{v_5, v_6, v_7, v_8\} = \{11\,1000, 10\,0100, 01\,0010, 00\,0001\}.$$

Note that all vectors of V_1 start with an element of W and end with four 0s. The vectors of V_2 start also with an element of W and have exactly one 1 among the last four components, but each in a different place.

We have to show that V is $(3, 5)$-admissible. Consider the table T defined as follows:

$$\begin{array}{ccccccc} z_6 & 1 & 1 & 0 & 0 & 0 & * \\ z_5 & 1 & 0 & 0 & 0 & * & 0 \\ z_4 & 0 & 1 & 0 & * & 0 & 0 \\ z_3 & 0 & 0 & * & 0 & 0 & 0 \\ z_2 & 0 & * & 0 & 0 & 0 & 0 \\ z_1 & * & 0 & 0 & 0 & 0 & 0. \end{array}$$

As we can easily see $d(z_i, v_j) \leqslant 2$ for all $i = 1, ..., 6$ and $j = 1, ..., 8$. Hence V is (3, 5)-admissible via T.

It remains to verify that V is not (4, 6)-admissible. Assume V is (4, 6)-admissible; then there exists a vector $z \in \{0, 1\}^6$ with $d(z, v_j) \leqslant 2$ for all $j = 1, ..., 8$. We try to find out how it looks.

If $z(1) = z(2) = 1$ then, because of $d(z, v_4) \leqslant 2$ and $d(z, v_8) \leqslant 2$, both $z(6) = 0$ and $z(6) = 1$ must hold. But this is not possible. By symmetry, the other possibilities for $z(1)$ and $z(2)$ also lead to a contradiction. Since there is no possibility to choose the first two components of z, such a z cannot exist. Thus, V is not (4, 6)-admissible.

Now we prove Proposition 2.4.

Proof. We choose a set $W \subseteq \{0, 1\}^d$ of cardinality $m + 1$. This is possible because $m < 2^d$. Let $W = \{w_1, ..., w_{m+1}\}$ and define $\bar{w} \in \{0, 1\}^d$ by $\bar{w}(j) = 1 - w(j)$. Let

$$\delta_{ij} = \begin{cases} 0 & \text{for} \quad i \neq j \\ 1 & \text{for} \quad i = j. \end{cases}$$

Now we construct $V \subseteq \{0, 1\}^{m+d+1}$:

$$V = \bigcup_{i=1}^{6} V_i,$$

with

$$V_1 = \{(\bar{w}_i(1), ..., \bar{w}_i(d), 0, ..., 0) : i = 1, ..., m+1\}$$

$$V_2 = \{(\bar{w}_i(1), ..., \bar{w}_i(d), \delta_{i,1}, ..., \delta_{i,m+1}) : i = 1, ..., m+1\}$$

$$V_3 = \{(w_i(1), ..., w_i(d), 0, ..., 0) : i = 1, ..., m+1\}$$

$$V_4 = \{(w_i(1), ..., w_i(d), 1, 0, ..., 0) : \bar{w}_i \notin W\}$$

$$V_5 = \{(w(1), ..., w(d), 0, ..., 0) : w, \bar{w} \notin W\}$$

$$V_6 = \{(w(1), ..., w(d), 1, 0, ..., 0) : w, \bar{w} \notin W\}.$$

We have to show that V is $(m, m+d)$-admissible but not $(m+1, m+d+1)$-admissible.

1. V is $(m, m+d)$-admissible. Define the table T to have exactly the vectors

$$(1) \quad \underbrace{0 \cdots 0}_{l-1} * \underbrace{0 \cdots 0}_{d-l} \underbrace{0 \cdots 0}_{m+1}, \qquad l = 1, ..., d$$

and

$$(2) \quad w_l(1) \cdots w_l(d) \underbrace{0 \cdots 0}_{l-1} * \underbrace{0 \cdots 0}_{m-l+1}, \qquad l = 1, ..., m+1.$$

Define

$$a(u, v) = |\{i\colon 1 \leqslant i \leqslant d \wedge u(i) \neq v(i) \wedge u(i) \neq * \wedge v(i) \neq *\}|$$

and $b(u, v)$ by

$$d(u, v) = a(u, v) + b(u, v).$$

We show that V is $(m, m+d)$-admissible via T; i.e., $d(y, v) \leqslant d$ for every $y \in T$ and $v \in V$. Let $y \in T$. If y is of the form (1), then, for every $v \in V$, we have $a(y, v) \leqslant d-1$ and $b(y, v) \leqslant 1$. If y is of the form (2), then we distinguish four cases.

1. $v \in V_1 \cup V_3 \cup V_5$. Then we have $a(y, v) \leqslant d$ and $b(y, v) = 0$.
2. $v \in V_2$. Then $b(y, v) \leqslant 1$ and $a(y, v) < d$ unless

$$v = \bar{w}_l(1) \cdots \bar{w}_l(d)\, \underbrace{0 \cdots 0}_{l-1}10 \cdots 0.$$

But in this case, $b(y, v) = 0$ as the only nonzero bit matches $*$.

3. $v \in V_4$, say $v = (w_i(1), ..., w_i(d), 0, ..., 0)$. If $a(y, v) = d$, then $y(j) = \bar{w}_l(j)$ for $j = 1, ..., d$. Hence $\bar{w}_i = w_l$ for some l, but this contradicts $w_l \in W$. Hence, for all $v \in V_4$ we have both $a(y, v) < d$ and $b(y, v) \leqslant 1$.

4. $v \in V_6$, say $v = (w(1), ..., w(d), 1, 0, ..., 0)$ with $w, \bar{w} \notin W$. If $a(y, v) = d$ we get the same contradiction as in Case 3. So we have again $a(y, v) < d$ and $b(y, v) \leqslant 1$.

2. We show that V is not $(m+1, m+1+d)$-admissible. To this end, let $y \in \{0, 1\}^{m+d+1}$ and define $u \in \{0, 1\}^d$ by $u(j) = y(j)$ for $1 \leqslant j \leqslant d$. We discuss the following three cases.

If $u \in W$ then there exist $v_1 \in V_1$ and $v_2 \in V_2$ such that $v_i(j) = 1 - u(j) = 1 - y(j)$ for $i = 1, 2$ and $j = 1, ..., d$. But $v_1 \neq v_2$ so that $d(y, v_i) \geqslant d+1$ for $i = 1$ or $i = 2$.

If $u \notin W$ and $\bar{u} \in W$ then there exist $v_3 \in V_3$ and $v_4 \in V_4$ such that $v_i(j) = \bar{u}(j) = 1 - y(j)$ for $i = 3, 4$ and $j = 1, ..., d$. But $v_3 \neq v_4$ so that $d(y, v_i) \geqslant d+1$ for $i = 3$ or $i = 4$.

If $u \notin W$ and $\bar{u} \notin W$ then there exist $v_5 \in V_5$ and $v_6 \in V_6$ such that $v_i(j) = \bar{u}(j) = 1 - y(j)$ for $i = 5, 6$ and $j = 1, ..., d$. But $v_5 \neq v_6$ so that $d(y, v_i) \geqslant d+1$ for $i = 5$ or $i = 6$.

Since there is no possibility of constructing a vector y without making more than d errors, V is not $(m+1, m+d+1)$-admissible. ∎

From the last proposition and the fact that $(m, m+d)\mathrm{P} \supseteq (m+1, m+d+1)\mathrm{P}$ we get $(m, m+d)\mathrm{P} \supsetneq (m+1, m+d+1)\mathrm{P}$ for $m < 2^d$. Hence we proved Theorem 1.1. It is very natural to ask what happens for $m \geqslant 2^d$. We conjecture

$$(m, m+d)\mathrm{P} = (m+1, m+d+1)\mathrm{P} \qquad \text{for} \quad m \geqslant 2^d.$$

But we succeeded only in proving the weaker result in Theorem 1.2.

In preparation for this proof let us recall Ramsey's theorem. For a set X, let $X^{[2]}=\{Y\subseteq X : |Y|=2\}$.

THEOREM 2.5 (Ramsey's theorem [Gra90]). *For all $n, q\in\mathbb{N}$ there exists $R(n,q)$ so that for all $N\geqslant R(n,q)$ and any mapping $\varphi\colon\{1,...,N\}^{[2]}\to\{1,...,q\}$ there exists a subset $M\subseteq\{1,...,N\}$ with $|M|\geqslant n$ such that $\varphi(M^{[2]})=\{\varphi(X)\colon X\in M^{[2]}\}$ is a singleton.*

PROPOSITION 2.6. *For every $d\in\mathbb{N}$ there exists a natural number $c(d)$ such that for $m\geqslant c(d)$ every $(m, m+d)$-admissible set is $(m+1, m+d+1)$-admissible.*

Proof. Let $d\in\mathbb{N}$ be given. We choose an m and assume that there exists a $V\subseteq\{0,1\}^{m+d+1}$ which is $(m, m+d)$-admissible, but not $(m+1, m+d+1)$-admissible. Using Ramsey's theorem we want to show that this is impossible for large m.

Let $z_1, z_2, ..., z_{m+d+1}$ be the rows of a table T with $z_i(i)=*$ such that V is $(m, m+d)$-admissible via T.

CLAIM 1. *For $i=1, ..., m+d+1$ there exist vectors $f_i, g_i\in V$ with*

$$f_i(i)=0,\qquad g_i(i)=1,\qquad\text{and}\qquad d(f_i, z_i)=d(g_i, z_i)=d.$$

Proof. If $*$ in z_i is replaced by 0 or 1 we get vectors $z_i^{(0)}$ and $z_i^{(1)}$, respectively. Since V is not $(m+1, m+d+1)$-admissible, there exist vectors $f_i, g_i\in V$ such that $d(f_i, z_i^{(1)})\geqslant d+1$ and $d(g_i, z_i^{(0)})\geqslant d+1$. On the other hand, V is $(m, m+d)$-admissible, and this means $d(f_i, z_j)\leqslant d$ and $d(g_i, z_j)\leqslant d$ for all i and j. From these statements we easily conclude the claim. ▮

Let us consider the mapping

$$\varphi\colon\{1, ..., m+d+1\}^{[2]}\to\{0, ..., 2d\}\times\{0, ..., d\}\times\{0, ..., d\}$$

defined by

$$\varphi(i,j)=(d(z_i, z_j), d(f_i, z_j), d(g_i, z_j)),\qquad\text{for}\quad i<j.$$

Let $q=(2d+1)(d+1)^2$ denote an upper bound of the cardinality of the range of φ and let $n=2^d+d+2$. Ramsey's theorem (Theorem 2.5) implies that for $m+d+1\geqslant R(n,q)$ there exists a subset $M\subseteq\{1, ..., m+d+1\}$ with $|M|=n$ such that the image of $M^{[2]}$ under φ is a singleton, say $\{(k, x, y)\}$.

First of all we note that $x=y=d$ is impossible. This follows from

CLAIM 2. $x\not\equiv y\pmod 2$.

Proof. For arbitrary $i, j\in M$, $i<j$, we consider the vectors f_i, g_i, z_j. Without loss of generality we assume z_i to be of the form shown as follows. Since f_i and g_i may differ in at most $2d$ components, $f_i^{-1}(1)\cap g_i^{-1}(1)\neq\emptyset$.

Without loss of generality this appears as follows:

$$\begin{array}{ll}
f_i\colon & \overbrace{1, \ldots, 1,\ 1, \ldots, 1,\ 1, \ldots, 1,\ 1, \ldots, 1}^{d},\ 0, \ldots, 0,\ 0, \ldots, 0,\ 0, \ldots, 0,\ 0, \ldots, 0,\ \overset{i}{0},\ \overset{j}{0},\ 0,\ 0, \ldots \\
g_i\colon & 0, \ldots, 0,\ 0, \ldots, 0,\ \underbrace{1, \ldots, 1,\ 1, \ldots, 1,\ 1, \ldots, 1,\ 1, \ldots, 1}_{d},\ 0, \ldots, 0,\ 0, \ldots, 0,\ 1,\ 0,\ 0, \ldots \\
 & \\
z_i\colon & 0, \ldots, 0,\ 0, \ldots, 0,\ 0, \ldots, 0,\ 0, \ldots, 0,\ 0, \ldots, 0,\ 0, \ldots, 0,\ 0, \ldots, 0,\ 0, \ldots, 0,\ *,\ 0,\ 0, \ldots \\
z_j\colon & \underbrace{1, \ldots, 1}_{a_1},\ 0, \ldots, 0,\ \underbrace{1, \ldots, 1}_{a_2},\ 0, \ldots, 0,\ \underbrace{1, \ldots, 1}_{b_1},\ 0, \ldots, 0,\ \underbrace{1, \ldots, 1}_{b_2},\ 0, \ldots, 0,\ c,\ *,\ 0, \ldots
\end{array}$$

The following equations must hold:

(i) $x = d(f_i, z_j) = d - (a_1 + a_2) + b_1 + b_2 + c$

(ii) $y = d(g_i, z_j) = d - (a_2 + b_1) + a_1 + b_2 + 1 - c$

Adding both equations we get

$$2d - 2a_2 + 2b_2 + 1 = x + y.$$

This shows that

$$x \not\equiv y \pmod 2. \quad \blacksquare$$

By symmetry we can assume $x < d$ and by choice of M, we have

$$n - 1 = \max_{i \in M} |\{ j \in M : j > i \wedge d(f_i, z_j) = x \}|.$$

We prove

CLAIM 3.

$$\max_{i \in M} |\{ j \in M : j > i \wedge d(f_i, z_j) = x \}| \leqslant 2^d + d.$$

Proof. We consider a vector f_i, the corresponding row z_i, and another row z_j such that $d(f_i, z_j) = x$ holds. Without loss of generality they have the following form:

$$\begin{array}{ll}
f_i\colon & \overbrace{1, \ldots, 1,\ 1, \ldots, 1}^{d},\ 0, \ldots, 0,\ 0, \ldots, 0,\ \overset{i}{0},\ 0, \ldots \\
z_i\colon & 0, \ldots, 0,\ 0, \ldots, 0,\ 0, \ldots, 0,\ 0, \ldots, 0,\ *,\ 0, \ldots \\
z_j\colon & \underbrace{1, \ldots, 1}_{a},\ 0, \ldots, 0,\ \underbrace{1, \ldots, 1}_{b},\ 0, \ldots, 0,\ c,\ 0, \ldots
\end{array}$$

How many rows z_j of this form can exist? There are at most d of them which can have the star $(*)$ under the first d components. If the star is not under the first d components, the following two equalities must hold:

(i) $a+b=d(z_i, z_j)=k$,

(ii) $d-a+b+c=d(f_i, z_j)=x$.

Since $x<d$, we get from (ii) $b-a+c<0$, hence $2a>k-1$ and $2b<k$. Thus a may be any value between $[(k-1)/2]$ and d. Since $k \in \{0, ..., 2d\}$, the maximal possible range for a is $\{1, ..., d\}$. Thus we have no more than 2^d possibilities to choose 1s under the first d components. Each of these possibilities gives rise to at most one z_j. To verify this, let us assume there exist two vectors z_j and z_k being equal on the first d components and having a 1s there, $1 \leqslant a \leqslant d$. Each of them has exactly b further 1s so that their hamming distance is bounded by $2b$. As $2b<k$, this contradicts the assumption $d(u, v)=k$ for all rows $u \neq v$. Thus, altogether we have at most 2^d+d vectors z_j with the property $d(f_i, z_j)=x$. ▮

By definition we have that $n=2^d+d+2$. But from Claim 3 we get $n \leqslant 2^d+d+1$ which implies a contradiction. We have thus proved that every $(m, m+d)$-admissible set is $(m+1, m+d+1)$-admissible for $m+d+1 \geqslant R(n, q)=R(2^d+d+2, (2d+1)(d+1)^2)$. ▮

From Proposition 2.6 and Theorem 2.3 we get Theorem 1.2.

3. MORE ERRORS MAY SAVE TIME

The idea of frequency computations is to get a faster solution of a complex problem by allowing some errors. That this is possible was proved by Kinber [Kinb75]. He showed that there exist arbitrarily complex functions in $(n-1, n)$P. But the proof is hard to understand. We get his result as a special case of a more general result (Theorem 1.3) which moreover gives some insight into the structure of the hierarchy of polynomial time frequency classes. We need some preparatory steps.

1. We consider step counting functions s of Turing machines which are strictly increasing in the following sense:

$$m<n \Rightarrow s(0^m)<s(0^n). \tag{1}$$

One easily verifies

LEMMA 3.1. *For every recursive function t there exists a step counting function s with property* (1) *majorizing t.*

2. For a step counting function s having property (1) we define inductively a sequence of words:

$$w_1=0$$

$$w_k=0^{s(|w_{k-1}|)}.$$

Let

$$\lambda(k) = |w_k|$$

and

$$S = \{w_1, w_2, ...\}.$$

LEMMA 3.2. *S is decidable in polynomial time.*

Proof. Let M be a Turing machine computing exactly $s(n)$ steps on inputs of the form 0^n. We describe a polynomial time decision procedure for S.

Inputs not having the form 0^k are refused. For inputs of the form 0^k we generate the sequence $w_1, w_2, ...$ word by word. If w_i is generated, M uses it as input, and in each step of M a unary counter for $s(|w_i|)$ is incremented. This yields w_{i+1}. The input is accepted if and only if there exists an i such that $w_i = 0^k$.

Because of the monotonicity of s, at most k initial words of 0^k can be generated. Hence the procedure needs no more than ck^2 steps for some $c > 0$. ∎

3. Let M'_1, M'_2, ... be an effective enumeration of all Turing machines having a clock t (t being a recursive function). Let the list M'_1, M'_2, ...be generated by an algorithm A.

Define recursively $M_1 = M'_1$, and if $M_i = M'_j$ then

$$M_{i+1} = \begin{cases} M_i & \text{if } A \text{ generates the list } (M'_1, ..., M'_j) \\ & \text{within } \lambda(i+1) \text{ steps} \\ M'_{j+1} & \text{if } A \text{ generates the list } (M'_1, ..., M'_j, M'_{j+1}) \\ & \text{within } \lambda(i+1) \text{ steps.} \end{cases}$$

So we can assume an effective enumeration $M_1, M_2, ...$ of all Turing machines working within time bound t such that M_i can be determined within $\lambda(i)$ steps given input i. Now we are ready to prove Theorem 1.3.

THEOREM 1.3. *Assume* $1 \leqslant m \leqslant n$, $1 \leqslant d$, $0 \leqslant k$ *for* $m, n, k, d \in \mathbb{N}$. *Let t be an arbitrary recursive function. Then*

$$(m+k, n+k+d)\ \mathrm{P} \backslash (m, n)\ \mathrm{TIME}(t) \neq \varnothing.$$

Proof. The proof is based on a wide space diagonalization argument. This is a common method which was used for instance by Amir and Gasarch in [AG88] and Kummer and Stephan in [KS91]. One of the first to use wide space diagonalization arguments was Ladner [Lad75].

For the given recursive function t choose a step counting function s according to Lemma 3.1 and using this s, define S and λ as above. In particular, we have

$$t(|w_k|) \leqslant |w_{k+1}|. \tag{2}$$

We introduce a partition $S=S_1 \cup S_2 \cup \cdots$ by defining sets S_j having exactly $n+k+d$ elements:

$$S_j=\{w_{(n+k+d)\,j-(n+k+d-1)}, ..., w_{(n+k+d)\,j}\}.$$

If T_j denotes the operator $T_j\colon (\Sigma^*)^n \to \{0, 1\}^n$ computed by M_j we define n values which T_j yields if it is applied to the n smallest elements of S_j:

$$\begin{aligned}(y_{(n+k+d)\,j-(n+k+d-1)}, ..., y_{(n+k+d)\,j-(k+d)})\\ =T_j(w_{(n+k+d)\,j-(n+k+d-1)}, ..., w_{(n+k+d)\,j-(k+d)}).\end{aligned}$$

Making use of these y_v we introduce the function $f\colon \Sigma^* \to \{0, 1\}$,

$$f(x)=\begin{cases} 1-y_v & \text{if } x=w_v \text{ and } v \text{ is of the form } v=(n+k+d)\,j-c \\ & \text{where } k+d \leqslant c \leqslant n+k+d-1 \\ 0, & \text{otherwise.}\end{cases}$$

We verify that $f \notin (m, n)\,\mathrm{TIME}(t)$, but $f \in (m+k, n+k+d)\,P$.

1. $f \notin (m, n)\,\mathrm{TIME}(t)$. Assume that M_j provides an (m, n)-computation of f. Apply T_j on the n-tuple of the n smallest elements of S_j. The first line of the definition of f shows that

If $n-m+d \leqslant n$, T_j makes $n-m+d>n-m$ errors,

If $n-m+d>n$, T_j makes $n>n-m$ errors, which contradicts the assumption. Hence, no M_j provides an (m, n)-computation of f.

2. $f \in (m+k, n+k+d)\,\mathrm{P}$. To show this we construct a polynomial time operator $T\colon (\Sigma^*)^{n+k+d} \to \{0, 1\}^{n+k+d}$ computing f with at most $n-m+d$ errors. T will be defined to be symmetric, i.e., its value does not depend on the order of its arguments. Therefore, we always assume the arguments $x_1, ..., x_{n+k+d}$ to be given in the lexicographic order $x_1<x_2< \cdots <x_{n+k+d}$. We now describe the construction of

$$(z_1, ..., z_{n+k+d})=T(x_1, ..., x_{n+k+d}).$$

For this purpose we define j_m to be the largest j s.t. $S_j \cap \{x_1, ..., x_{n+k+d}\} \neq \varnothing$. Then we put

$$z_i=\begin{cases} 0 & \text{if } \; x_i \in S_{j_m} \wedge |S_{j_m} \cap \{x_1, ..., x_{n+k+d}\}| < n+k+d \\ f(x_i) & \text{otherwise.}\end{cases}$$

We notice that $z_i=f(x_i)$ with at most $n-m+d$ exceptions. These exceptions may occur if $x_i \in S_{j_m} \wedge |S_{j_m} \cap \{x_1, ..., x_{n+k+d}\}| < n+k+d$, because in this case we have $z_i=0$ and $f(x_i)=1$ by the first line of the definition of f. But by the definition of f this is possible for no more than $n-m+d$ arguments. This shows that T is an $(m+k, n+k+d)$-computation of f.

Finally, we have to verify that T is computable within polynomial time with respect to $|x_{n+k+d}|$ for any given tuple $(x_1, ..., x_{n+k+d})$. At first, we test which of the x_i are in S. For those that are, we determine j and l with $x_i = w_{(n+k+d)\,j-l}$ for $0 \leqslant l \leqslant n+k+d-1$. By our Lemma 3.2 this needs polynomial time in $|x_{n+k+d}|$.

Now we compute z_i for $i = 1, ..., n+k+d$.

If $x_i \in S_{j_m}$ and $|S_{j_m} \cap \{x_1, ..., x_{n+k+d}\}| = n+k+d$, then we have to compute

$$T_{j_m}(w_{(n+k+d)\,j_m-(n+k+d-1)}, ..., w_{(n+k+d)\,j_m-(k+d)}). \tag{3}$$

If x_i belongs to the n smallest elements of some S_j, $j \neq j_m$, then we have to compute

$$T_j(w_{(n+k+d)\,j-(n+k+d-1)}, ..., w_{(n+k+d)\,j-(k+d)}). \tag{4}$$

In all remaining cases $z_i = 0$, which requires only constant time. To compute (3) we have to

- determine M_{j_m} and
- apply M_{j_m} to $(w_{(n+k+d)\,j_m-(n+k+d-1)}, ..., w_{(n+k+d)\,j_m-(k+d)})$.

The former requires

$$\lambda(j_m) = |w_{j_m}| \leqslant |w_{(n+k+d)\,j_m-(n+k+d-1)}| \leqslant |x_{n+k+d}|$$

steps; the latter requires

$$t(|w_{(n+k+d)\,j_m-(k+d)}|) \leqslant |w_{(n+k+d)\,j_m-(k+d-1)}| \leqslant |w_{(n+k+d)\,j_m}| = |x_{n+k+d}|$$

steps. Notice that $x_{n+k+d} = w_{(n+k+d)\,j_m}$ since $x_{n+k+d} \in S_{j_m}$. The first inequality follows from (2).

The argument for (4) is nearly the same. By (2) we get

$$t(|w_{(n+k+d)\,j-(k+d)}|) \leqslant |w_{(n+k+d)\,j-(k+d-1)}| \leqslant |x_{n+k+d}|.$$

Here it is important that $w_{(n+k+d)\,j-(k+d-1)}$ or some larger element of S belongs to $(x_1, ..., x_{n+k+d})$ so that $|x_{n+k+d}|$ majorizes the computation time. Hence, we can perform all steps within polynomial time with respect to $|x_{n+k+d}|$. ∎

For $m = n$, $k = 0$ and $d = 1$ we get

COROLLARY 3.3 [Kinb75]. *For arbitrarily large recursive t there exist functions*

$$f \in (n-1, n)\ \mathrm{P} \backslash \mathrm{TIME}(t).$$

Remark. Kinber [Kinb75] proves that P can be replaced by TIME(id).

If the set S is defined by

$$w_1 = 0$$
$$w_{i+1} = 0^{2^{|wi|}},$$

Lemma 3.2 remains valid, and we can prove in exactly the same way that Theorem 1.3 remains valid if TIME(t) is replaced by P. This gives several important corollaries.

The following three corollaries follow already from Theorem 2.3. We mention them here once more explicitly to get Fig. 1 as complete as possible. From $\mathrm{P} \subseteq (2, 3)\,\mathrm{P}$ and $(2, 3)\,\mathrm{P} \backslash \mathrm{P} \neq \emptyset$ (Theorem 1.3) we get

Corollary 3.4.

$$\mathrm{P} \subsetneq (2, 3)\,\mathrm{P}.$$

For $k = 0$ and $d = 1$ we get $(m, n+1)\,P \backslash (m, n)\,P \neq \emptyset$ which shows, that statement (1) of Theorem 2.1 is a strict inclusion:

Corollary 3.5.

$$(m, n)\,\mathrm{P} \subsetneq (m, n+1)\,\mathrm{P}.$$

Finally, we see that no frequency class is contained in another frequency class with a smaller number of errors.

Corollary 3.6. For $d \geqslant 1$ and $k \geqslant 0$,

$$(m+k, n+k+d)\,\mathrm{P} \nsubseteq (m, n)\,P.$$

In order to summarize the results obtained on the hierarchy of the polynomial time frequency classes we now mention some more results.

Theorem 3.7.

1. [Kin75]. $(2, 3)\,\mathrm{P} = (2+k, 3+k)\,\mathrm{P}$ for $k \geqslant 1$.
2. [Hin94]. $(4, 6)\,\mathrm{P} = (4+k, 6+k)\,\mathrm{P}$ for $k \geqslant 1$.
3. [KS95]. $(2, 4)\,\mathrm{P} \subseteq (3, 6)\,\mathrm{P}$.
4. [KS95]. For all $m \in \mathbb{N}$ there exists a number h_0 such that for all $h \geqslant h_0$, $(m, 2m)\,\mathrm{P} \subseteq (h, 2h)\,\mathrm{P}$ holds.
5. [KS95]. For $m < h \in \mathbb{N}$, $(m, 2m+1)\,\mathrm{P} \backslash (h, 2h+1)\,\mathrm{P} \neq \emptyset$.

The proof of number 3 can be generalized so that we get the following result:

Theorem 3.8. *For* $d, k \in \mathbb{N}$, $d \geqslant 2$, $k \geqslant 0$, $(d+k, 2d+k)\,\mathrm{P} \subseteq (d+k+1, 3d+k)\,\mathrm{P}$.

Some other inclusions can be proved similarly; e.g., one can show that

- $(2, 5)\,\mathrm{P} \subseteq (3, 8)\,\mathrm{P}$
- $(4, 7)\,\mathrm{P} \subseteq (6, 11)\,\mathrm{P}$.

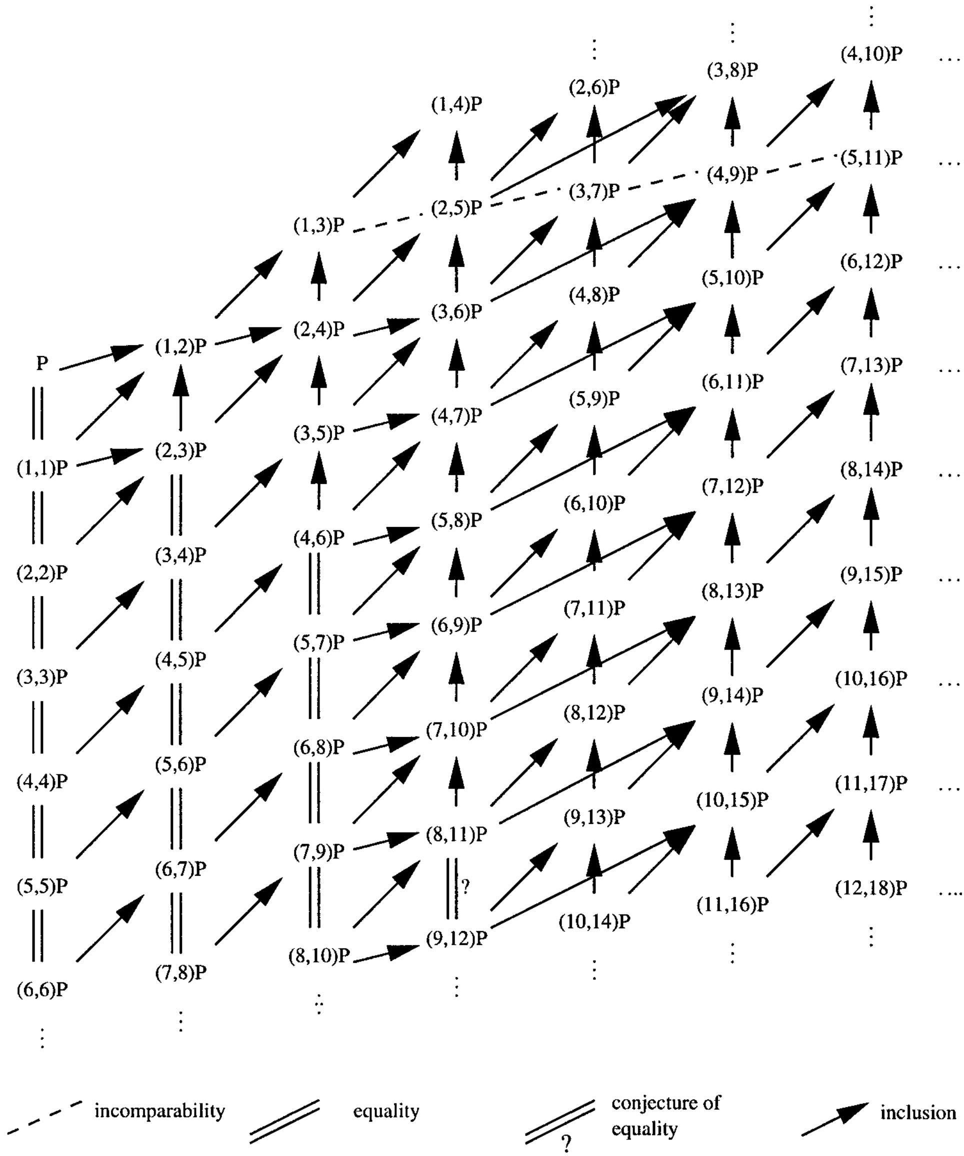

FIG 1. The hierarchy of the frequency classes under polynomial time bounds. Note that, for clarity reasons, not all known results mentioned in the paper are incluced.

4. SOME CONNECTIONS BETWEEN SELECTIVITY CLASSES AND FREQUENCY CLASSES

Theorem 1.3 shows that there exist arbitrarily complex problems which are frequency computable in polynomial time. A natural question is whether this is possible for NP-complete problems.

It has been proved [KS95], [Ogi94] that the (m, n)-computability of SAT implies $P = NP$. (We discuss this result at the end of this section.) Selman studied the concept of semirecursive sets introduced by Jockusch [Joc68] in a polynomial time setting and proved that if SAT is P-selective, then $P = NP$. Are these two collapse results related to each other? To answer this question we compare the P-selective sets with the frequency computable sets.

According to [Sel79] we define

DEFINITION 4.1. The set A is called P-selective iff there exists a function f (a selector for A) computable in polynomial time such that

1. $f(x, y) \in \{x, y\}$,
2. $x \in A \vee y \in A \Rightarrow f(x, y) \in A$.

Let $P - \mathrm{Sel}$ be the set of all P-selective sets.

In analogy to the statement that any semirecursive set is (1, 2)-recursive (see [KS91]) we have

Fact 4.2. $P - \mathrm{Sel} \subseteq (1, 2)\,P$.

We get this result as a special case of Theorem 4.6.

Hemaspaandra *et al.* [HJRW95] introduced a more general version of P-selectivity, the multi-selectivity. We give their definition in a slightly modified form:

DEFINITION 4.3. *$S(k)$ is the class of all sets $L \subseteq \Sigma^*$ for which there exists a function $f \in \mathrm{FP}$ (an $S(k)$-selector for L) such that for each $n \geqslant 1$ and any mutually distinct input strings $y_1, ..., y_n$,*

1. $f(y_1, ..., y_n) \in \{y_1, ..., y_n\}$ and
2. $|L \cap \{y_1, ..., y_n\}| \geqslant k \Rightarrow f(y_1, ..., y_n) \in L$.

The following results were proved in the paper mentioned above.

Fact 4.4.

1. $S(k) \subset S(k+1)$ for each $k \geqslant 1$,
2. $S(1) = P - \mathrm{Sel}$.

We now investigate the relationship between the classes $(m, n)\,P$ and $S(k)$.

THEOREM 4.5. *For all $k \geqslant 1, m, n$ $(m < n)$*

$$(m, n)\,P \nsubseteq S(k).$$

Proof. Because of Kinber's result (Theorem 3.7) it is sufficient to prove $(2, 3)\,P \backslash S(k) \neq \varnothing$. (Note that $(2, 3)\,P$ is the smallest polynomial time frequency class.) We consider the set

$$S = \{w_1, w_2, ...\},$$

where

$$w_1 = 0$$
$$w_{n+1} = 0^{2^{|w_n|}}.$$

Let $M_1, M_2, ...$ be an effective enumeration of all polynomial time Turing machines computing all possible polynomial time selectors $f_1, f_2, ...$. By diagonalizing against all polynomial time computable selector functions we construct a set $A \in (2, 3)$ P such that $A \notin S(k)$ for all $k \geqslant 1$. Let

$$S = S_1 \cup S_2 \cup \cdots,$$

where

$$S_j = \{w_{(k+2)\,j-(k+1)}, w_{(k+2)\,j-k}, ..., w_{(k+2)\,j}\}.$$

We define A by

1. $x \in A$, if $x \notin S$ or there exists a j such that $x = w_{(k+2)\,j}$.
2. $w_{(k+2)\,j-h} \notin A \wedge w_{(k+2)\,j-i} \in A$ for $h, i = 1, ..., (k+1)$, $i \neq h$ $\Leftrightarrow f_j(w_{(k+2)\,j-(k+1)}, w_{(k+2)\,j-k}, ..., w_{(k+2)\,j-1}) = w_{(k+2)\,j-h}$.

It follows immediately from the definition of A that no f_j can be an $S(k)$-selector of A. Hence, $A \notin S(k)$.
But $A \in (2, 3)$ P. To show this, we consider the operator

$$T(x_1, x_2, x_3) = (u_1, u_2, u_3),$$

which is defined as follows:

1. If $x_i \notin S$ or $\bigvee_j x_i = w_{(k+2)\,j}$, then $u_i = 1$.
2. Let $j_{\max}$ be the largest j such that $S_j \cap \{x_1, x_2, x_3\} \neq \emptyset$.
3. If $x_i = w_{(k+2)\,j-l}$, $l = 1, ..., (k+1)$ and $j < j_{\max}$, compute

$$v = f_j(w_{(k+2)\,j-(k+1)}, ..., w_{(k+2)\,j-1})$$

and let

$$u_i = \begin{cases} 0 & \text{if} \quad v = w_{(k+2)\,j-l} \\ 1 & \text{otherwise.} \end{cases}$$

4. We now consider the case where $|S_{j_{\max}} \cap \{x_1, x_2, x_3\}| = 3$.

(a) If $(x_1, x_2, x_3) = (w_{(k+2)\,j_{\max}-l_1}, w_{(k+2)\,j_{\max}-l_2}, w_{(k+2)\,j_{\max}-l_3})$, $1 \leqslant l_1, l_2, l_3 \leqslant (k+1)$, then $u_1 = u_2 = u_3 = 1$.

(b) If $x_3 = w_{(k+2)\,j_{\max}}$ and $x_i = w_{(k+2)\,j_{\max}-l_i}$, $i \in \{1, 2\}$, $1 \leqslant l_i \leqslant (k+1)$, then compute

$$v = f_{j_{\max}}(w_{(k+2)\,j_{\max}-(k+1)}, ..., w_{(k+2)\,j_{\max}-1})$$

and let for $i \in \{1, 2\}$

$$u_i = \begin{cases} 0 & v = w_{(k+2)\, j_{\max} - l_i} \\ 1 & \text{otherwise} \end{cases}$$

and

$$u_3 = 1.$$

5. If $1 \leqslant |S_{j_{\max}} \cap \{x_1, x_2, x_3\}| \leqslant 2$, then $u_i = 1$ if $x_i \in S_{j_{\max}}$.

Clearly, this T (2, 3)-computes c_A: Only in cases 4(a) and 5 can errors occur, actually at most one error. T is computable in polynomial time. The argument is practically the same as in the proof of Theorem 1.3. ▮

THEOREM 4.6. *For any $k \geqslant 1$ and any $m \geqslant 1$,*

$$S(k) \subseteq (m, 2m+k-1)\text{ P}.$$

Proof. Let $A \in S(k)$. We have to define a polynomial time operator T which $(m, 2m+k-1)$-computes A.

Let f be an $S(k)$-selector function. We consider a $(2m+k-1)$-tuple $(x_1, ..., x_{2m+k-1})$ and assume w.l.o.g. that the indices are chosen so that

$$f(x_1, ..., x_{2m+k-1}) = x_{2m+k-1}$$

$$f(x_1, ..., x_{2m+k-2}) = x_{2m+k-2}$$

$$\vdots$$

$$f(x_1, ..., x_{m+k}) = x_{m+k}.$$

Now, we define

$$T(x_1, ..., x_{2m+k-1}) = \underbrace{0...0}_{m+k-1}\underbrace{1...1}_{m}.$$

The claim is that this T computes A with at most $m+k-1$ errors. Let $i = |A \cap \{x_1, ..., x_{2m+k-1}\}|$.

Case 1. $i \geqslant m+k-1$.
In this case, all 1s are correct, i.e., we have at least m correct outputs.

Case 2. $k \leqslant i \leqslant m+k-2$.
In this case, the last $i-k+1$ 1s are correct, and the remaining 1s need not be correct, because an $S(k)$-selector applied to a set with less than k elements from A need not output an element from A. Therefore, $k-1$ 0s could be wrong. Hence, we have $m+k-1-(k-1) = m$ 0s which are correct. So, altogether, we have $i-k+1+m \geqslant m+1$ correct outputs.

Case 3. $i \leqslant k-1$.

None of the 1s need be correct. Furthermore, i of the 0s could be false. There remain $m+k-1-i \geqslant m$ correct 0s.

T is computable in polynomial time because $f \in \mathrm{FP}$. ▮

The next two theorems show that Theorem 4.6 gives the optimal embedding of the $S(k)$-classes into the hierarchy of the polynomial time bounded frequency classes.

THEOREM 4.7. *For all* $k \geqslant 1$

$$S(k) \nsubseteq (1, k)\,\mathrm{P}.$$

Proof. Let $M_1, M_2, \ldots$ be an effective enumeration of all polynomial time Turing machines computing all possible polynomial time operators $T_1, T_2, \ldots$. For fixed k we now construct a set A which is in $S(k)$ but not in $(1, k)\,\mathrm{P}$ by diagonalizing against all polynomial time operators $T_1, T_2, \ldots$. We make use of the set $S = S_1 \cup S_2 \cup \cdots$ introduced in the proof of the previous theorem with the only difference that

$$S_j = \{w_{(k+1)j-k}, w_{(k+1)j-(k-1)}, \ldots, w_{(k+1)j}\}.$$

Then A is defined by

1. $x \in A$, if $x \notin S$ or $\bigvee_j x = w_{(k+1)j}$.
2. Let $T_j(w_{(k+1)j-k}, \ldots, w_{(k+1)j-1}) = (u_{(k+1)j-k}, \ldots, u_{(k+1)j-1})$. Then $\bigwedge_j \bigwedge_{l=1}^{k} w_{(k+1)j-l} \in A \Leftrightarrow u_{(k+1)j-l} = 0$.

From this definition of A it follows directly that $A \notin (1, k)\,\mathrm{P}$, because each operator T_j makes k errors when it is applied to

$$(x_1, x_2, \ldots, x_k) = (w_{(k+1)j-k}, w_{(k+1)j-(k-1)}, \ldots, w_{(k+1)j-1}).$$

Now we give an $S(k)$-selector for A. For all $n \geqslant 1$ and input strings $x_1, x_2, \ldots, x_n$ with mutually distinct x_i and $x_1 < x_2 < \cdots < x_n$ we define f as follows:

1. If among the arguments there exists an element not belonging to S, then $f(x_1, x_2, \ldots, x_n) = x_i$, if i is the smallest index j s.t. $x_j \notin S$.

2. Otherwise, if among the arguments there exists some $w_{(k+1)j}$, then $f(x_1, x_2, \ldots, x_n) = x_i$, if i is the smallest l s.t. $\bigvee_j x_l = w_{(k+1)j}$.

3. If none of the preceding cases applies we have the case $\bigwedge_{i=1}^{n} \bigvee_j \bigvee_{l=1}^{k} (x_i \in S_j \wedge x_i = w_{(k+1)j-l})$. If $n < k$, let $f(x_1, \ldots, x_n) = x_1$. If $n \geqslant k$, we proceed as follows. Making use of the second line of the definition of A we check whether there exists at least one element among $x_1, x_2, \ldots, x_{n-k}$ which belongs to A. If there exists such an element, let x_i be the smallest of them and let

$$f(x_1, \ldots, x_n) = x_i.$$

Otherwise, put $f(x_1, \ldots, x_n) = x_{n-k+1}$.

Finally, we verify that f is an $S(k)$-selector for the set A. We have to show that f outputs an element from the input string $\{x_1, ..., x_n\}$ which is in A, if there are at least k elements from the input string in A. In the cases 1 and 2 there is at least one element which is in A. This becomes the output for f. So we have a correct output, if there are at least k elements in A.

In Case 3 we first test whether there is an element of $\{x_1, ..., x_{n-k}\}$ in A for sure. If we find one, this is a correct output of f. If not we can output an arbitrary $x_i \in \{x_{n-k+1}, ..., x_n\}$, because if there are k elements from $\{x_1, ..., x_n\}$ in A, then these must be the elements $x_{n-k+1}, ..., x_n$.

f is computable in polynomial time with respect to $|x_n|$. The test whether an x_i lies in S and if yes in which S_j and whether it is the largest element in S_j is possible in polynomial time. In Case 3 we have to compute at most $n-k$ polynomial functions T_j, but only at arguments which are all smaller than x_n. So the worst case which can occur is that $x_{n-k}, x_{n-k+1}, ..., x_n$ are in $S_{j_{\max}}$. Then we have to compute $T_{j_{\max}}(x_{n-k}, ..., x_{n-1})$ which is polynomial in $|x_n|$. ∎

THEOREM 4.8. *For all* $k, m > 1$,

$$S(k) \nsubseteq (m, 2m+k-2)\text{ P}.$$

Proof. Our goal is to construct a set A such that $A \notin (m, 2m+k-2)$ P but having an $S(k)$-selector $f \in \text{FP}$. We use the set $S = S_1 \cup S_2 \cup \cdots$ with $S_j = \{w_{(2m+k-2)j-(2m+k-3)}, ..., w_{(2m+k-2)j}\}$. We make use of the shorthands $Y_j = T_j(w_{(2m+k-2)j-(2m+k-3)}, ..., w_{(2m+k-2)j}) = (y_{(2m+k-3)}, ..., y_0)$ and $d = (2m+k-2) - m = m+k-2$. The set A is defined by:

1. $x \in A$, if $x \notin S$ or $\bigvee_j x = w_{(2m+k-2)j}$.

Otherwise,

2. $S_j \subseteq A$, if $|\{i: y_i = 0\}| \geqslant d+1$.
3. $S_j \cap A = \emptyset$, if $|\{i: y_i = 1\}| \geqslant d+1$.

Assume that none of the cases 2 or 3 applies.

4. If $|\{i: y_i = 0\}| = d - l$, $0 \leqslant l \leqslant [d/2]$, then let $i_0, ..., i_l$ be the $(l+1)$ largest indices s.t. $y_{i_0} = y_{i_1} = \cdots = y_{i_l} = 1$. In this case $S_j \cap A = S_j \backslash \{w_{(2m+k-2)j-i_0}, ..., w_{(2m+k-2)j-i_l}\}$.

If Case 4 does not apply, then

5. If $|\{i: y_i = 1\}| = d - l$, $0 \leqslant l < [d/2]$, then let $i_0, ..., i_l$ the $(l+1)$-smallest indices s.t. $y_{i_0} = y_{i_1} = \cdots = y_{i_l} = 0$. In this case $S_j \cap A = \{w_{(2m+k-2)j-i_0}, ..., w_{(2m+k-2)j-i_l}\}$.

Notice that both $[d/2] \leqslant |\{i: y_i = 0\}| \leqslant d$ and $[d/2] \leqslant |\{i: y_i = 1| \leqslant d$ can happen simultaneously, but then Case 4 has priority over Case 5.

It is easy to verify that $A \notin (m, 2m+k-2)$ P. Indeed, assume that T_j $(m, 2m+k-2)$-computes A. By definition of A, T_j makes at least $d+1$ errors on S_j. Now we give an $S(k)$-selector f for A.

For a given set X with at least $k+1$ elements, we define $f(X)$ as follows. First, we determine in polynomial time which of the elements of X do not belong to S or for which there exists a j such that $x = w_{(2m+k-2)j}$. If there exist such elements, define $f(X)$ as the smallest of them. Otherwise, we determine (also in polynomial time) to which S_j the elements of X belong. For all $x \notin S_{j_{\max}}$, we exactly compute whether they belong to A or not. If there exist such elements belonging to A, then let $f(X)$ be the smallest of them. Otherwise, define $f(X)$ as the smallest element of $S_{j_{\max}}$.

This function f is an $S(k)$-selector for A. To verify this we distinguish several cases. We always assume that X contains at least k elements from A, since otherwise there is nothing to be shown.

Case 1. X contains an element $x \notin S_{j_{\max}}$ from A. Then the smallest element of this kind is chosen by the function f, i.e. f behaves correctly.

Case 2. No element from $X \backslash S_{j_{\max}}$ belongs to A. We consider the following subcases where now $Y_{j_{\max}} = \{y_{2m+k-3}, ..., y_0\}$.

1. If $|\{i: y_i = 0\}| \geqslant d+1$, then $S_{j_{\max}} \subseteq A$ and therefore the smallest element of $S_{j_{\max}}$ belongs to A.

2. If $|\{i: y_i = 0\}| = d - l$, $0 \leqslant l \leqslant [d/2]$, then if for the smallest element of $S_{j_{\max}}$ the corresponding y_i equals 0 it belongs to A by definition. Otherwise, according to Case 4 of the definition of A at least this element is in A, because of

$$|\{i: y_i = 1\}| = (2m+k-2) - |\{i: y_i = 0\}| = 2m+k-2-(d-l) = m+l > l+1.$$

3. If $|\{i: y_i = 0\}| < [d/2]$, we distinguish two subcases.

(a) $m < [d/2]$

In this case, $2m+k-2-[d/2] = m+d-[d/2] < |\{i: y_i = 1\}|$. According to Case 5 of the definition of A we represent $|\{i: y_i = 1\}|$ as $d-l$ and get $m+d-[d/2] < d-l$, which yields $l+1 < k/2$. But this means that $S_{j_{\max}} \cap A$ contains less than k elements. Thus we have no requirement to f.

(b) $[d/2] \leqslant m$ In this case we get $|\{i: y_i = 1\}| > m+d-[d/2] \geqslant d$. By Case 2 of the definition of A we get $S_{j_{\max}} \cap A = \emptyset$. Hence, again, f needs not to satisfy any condition.

Finally, we see that f is computable in polynomial time in the same way as in the preceding proofs. ∎

The next theorem shows that, for a frequency greater than 1/2, none of the selectivity classes is included in any of the polynomial time frequency classes. (See Fig. 2)

THEOREM 4.9. *For all m, n $m/n > 1/2$,*

$$S(1) \nsubseteq (m, n)\,\mathrm{P}.$$

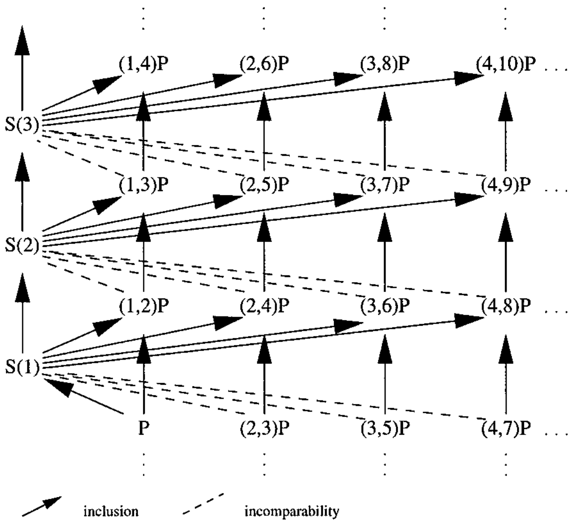

FIG 2. Relationship between selectivity and frequency classes.

Proof. To keep the proof easy we restrict ourselves to the case $(m, n) = (2, 3)$ and show $\mathrm{P} - \mathrm{Sel} \not\subseteq (2, 3)\,\mathrm{P}$. The general case can be treated similarly.

Our goal is to construct a set A such that $A \notin (2, 3)\,\mathrm{P}$ but having a selector $f \in \mathrm{FP}$. We make use of the set $S = S_1 \cup S_2 \cup \cdots$ with $S_j = \{w_{3j-2}, w_{3j-1}, w_{3j}\}$.

The construction of A makes use of an auxiliary function α, defined on S.

$$\alpha(j) = \begin{cases} 0 & \text{if the majority of the values of } T_j(w_{3j-2}, w_{3j-1}, w_{3j}) \\ & \text{equals 1} \\ 1 & \text{otherwise.} \end{cases}$$

The set A is defined as follows:

$$x \in A \Leftrightarrow \bigvee_j (x \in S_j \wedge \alpha(j) = 1).$$

It is easy to verify that $A \notin (2, 3)\,\mathrm{P}$. Assume T_j (2, 3)-computes A. If $\alpha(j) = 0$, then at least two components of $T_j(w_{3j-2}, w_{3j-1}, w_{3j})$ have the value 1. But $A \cap S_j = \varnothing$. Hence, T_j makes at least two errors on S_j. If $\alpha(j) = 1$, then at least two components of $T_j(w_{3j-2}, w_{3j-1}, w_{3j})$ have value 0. As $S_j \subseteq A$, T_j makes at least two errors on S_j.

Now we give a selector f for A:

$$f(x, y) = f(y, x) \qquad \text{for all} \quad x, y$$

1. $f(x, y) = \min(x, y)$ for $x, y \notin S$
2. $f(x, y) = \min(x, y)$ for $x \in S_j \wedge y \in S_j$
3. $f(x, y) = x$ if $y \notin S \wedge x \in S$
4. $f(x, y) = y$ if $x \in S_i \wedge y \in S_j \wedge i < j \wedge \alpha(i) = 0$
5. $f(x, y) = x$ if $x \in S_i \wedge y \in S_j \wedge i < j \wedge \alpha(i) = 1$.

We verify that f is a selector function for A. Let x, y be given. At first we find out which of the cases of the definition applies. If both x, y are not in S, then we can output an arbitrary string for instance the minimum of x, y. If both are in the same S_j then we can output one of them, because either both are in A or both are not. If only one of them is in S, we output this, because if one element of x, y is in A, then only this can be that one which is in S. If $x \in S_i$ and $y \in S_j$ and $i < j$, then we output x if $A \subseteq S_i$ and y otherwise. In the latter case, y is the only value which possibly can belong to A.

Finally, we show $f \in \mathrm{FP}$. Let x, y be given. As $S \in \mathrm{P}$ one can find out in polynomial time which of the cases of the definition of f applies for any given x, y. In the first three cases the value of f can readily be written down. In cases 4 and 5 we have to know $\alpha(i)$. To get it, we have to compute T_i on S_i which can be done in time $|y|$. ▮

We come back to the topic discussed at the beginning of this section. From [HJRW95] we know

THEOREM 4.10.

$$\bigcup_{k=1}^{\infty} S(k) \subseteq \mathrm{P/poly}.$$

The special case $\mathrm{P} - \mathrm{Sel} \subseteq \mathrm{P/poly}$ was already known to Ko [Ko83].

From Theorem 4.10 it follows that for no k can NP have $\leqslant_t^p$-complete sets which are k-selective, unless by a well known result of Karp and Lipton the polynomial hierarchy collapses on its second level.

Selman's result is stronger as it concerns a collapse $\mathrm{P} = \mathrm{NP}$. The result mentioned above that the polynomial time frequency computability of SAT implies $\mathrm{P} = \mathrm{NP}$ does not follow from Selman's result because of Theorem 4.5. Moreover, it would still be compatible with Theorems 4.9 and 4.10 that NP-complete problems would have polynomial time frequency computations. However, NP-complete problems admit polynomial time frequency computations only if the polynomial hierarchy collapses on its second level. This follows from

THEOREM 4.11 [ABG90, BKS95].

$$\bigcup_{k=2}^{\infty} (1, k)\,\mathrm{P} \subseteq \mathrm{P/poly}.$$

Stronger collapse results follow from

THEOREM 4.12 [Ogi94, BKS95]. *If* SAT *is btt-reducible in polynomial time to a set from* $\bigcup_{k=2}^{\infty} (1, k)\,\mathrm{P}$, *then* $\mathrm{P} = \mathrm{NP}$.

This theorem implies both collapse results mentioned at the beginning of this section.

CONCLUSION

We have proved some explicit solutions for important inclusion problems for polynomial time bounded frequency classes. But we still do not know what happens for

$(m, m+d)$ P and $(m+1, m+d+1)$ P if $2^d \leqslant m \leqslant R(2^d+d+2,\ (2d+1)(d+1)^d) - (d+1)$. We conjecture equality. But a general proof seems to be a very difficult combinatorial issue.

Nevertheless, we gained insight into the hierarchy of frequency classes under polynomial time restrictions.

ACKNOWLEDGMENTS

We thank two anonymous referees for corrections and comments, which helped to simplify the proof of Theorem 2.6.

Received February 13, 1997; final manuscript received July 7, 1997

REFERENCES

[ABG90] Amir, A., Beigel, R., and Gasarch, W. I. (1990), Some connections between bounded query classes and non-uniform complexity, *in* "Proc. 5th Conference on Structure in Complexity Theory," pp. 232–243, IEEE Press, New York.

[AG88] Amir, A., and Gasarch, W. I. (1988), Polynomial terse sets, *Inform. and Comput.* **77**, 37–56.

[BKS95] Beigel, R., Kummer, M, and Stephan, F. (1995), Approximable sets, *Inform. and Comput.* **120**, 304–314.

[Deg81] Degtev, A. N. (1981), On (m, n)-computable sets, *in* "Algebraic Systems" (D. I. Moldavanskij, Eds.), pp. 88–99, Ivanova Gos. Univ. [In Russian].

[Gra90] Graham, R. L., Rothschild, B. L., and Spencer, J. H. (1990), "Ramsey Theory", 2nd ed., Wiley, New York.

[HJRW95] Hemaspaandra, L. A., Jiang, Z., Rothe, J., and Watanabe, O. (1995), "Multi-selectivity and Complecity-Lowering Joins," Technical Report No. 568, Department of Computer Science, University of Rochester.

[Hin94] Hinrichs, M. (1994), "Häufigkeitsberechnungen," Diplomarbeit, Friedrich-Schiller-Universität Jena.

[HN93] Hoene, A., and Nickelsen, A. (1993), Counting, selecting and sorting by querybounded machines, *in* "STACS 93," Lecture Notes in Computer Science, Vol. 665, pp. 196–205, Springer-Verlag, Berlin/New York.

[Joc68] Jockusch, C. (1968), Semirecursive sets and positive reducibility, *Trans. Amer. Math. Soc.* **131** (2), 420–436.

[Kin75] Kinber, E. B. (1975), "Frequency-Computable Functions and Frequency-Enumerable Sets," Candidate dissertation, Riga. In Russian.

[Kinb75] Kinber, E. B. (1983), On frequency real-time computations, *in* "Teoriya Algorithmov i Programm" (Ya. M. Barzdin, Ed.), Latv. Valst. Gos. Univ., Vol. 2, pp. 174–182. In Russian.

[Ko83] Ko, K. I. (1983), On self-reducibility and weak P-selectivity, *J. Comput. System Sci.* **26**, 209–221.

[KS91] Kummer, M., and Stephan, F. (1991), "Some Aspects of Frequency Computation," Technical Report No. 21/91, Fakultät für Informatik, Universität Karlsruhe.

[KS95] Kummer, M., and Stephan, F. (1995), The power of frequency computation, *in* "FCT '95," Lecture Notes in Computer Sciences, Vol. 965, pp. 323–332, Springer-Verlag, Berlin/New York.

[Lad75] Ladner, R. E. (1975), On the structure of polynomial time reducibility, *J. Assoc. Comput. Mach.* **22**, 155–171.

[McN61] McNaughton, R. (1961), The theory of automata, *Adv. Comput.* **2**, 379–421.

[McNic95] McNicholl, T. (1995), "The Inclusion Problem for Generalized Frequency Classes," Ph.D. Thesis, Dept. of Mathematics, George Washington University, Washington DC.

[Nic97] Nickelsen, A. (1997), On polynomially $\mathscr{D}$-verbose sets, *in* "STACS '97," Lecture Notes in Computer Science, Vol. 1200, pp. 307–318, Springer-Verlag, Berlin/New York.

[Ogi94] Ogihara, M. (1994), Polynomial-time membership comparable sets, *in* "Proc. 9th Conference on Structure in Complexity Theory," pp. 2–11, IEEE Press, New York.

[Ros60] Rose, G. F. (1960), An extended notion of computability, *in* "Abstracts, International Congress for Logic, Methodology, and Phil. of Science, Stanford, California, 14."

[Sel79] Selman, A. (1979), *P*-selective sets, tally languages, and the behavior of polynomial time reducibilities on *NP*, *Math. Systems Theory* **13**, 55–65.

[Tra63] Trakhtenbrot, B. A. (1963), On frequency computation of functions, *Algebra i Logika* **2**, 25–32. In Russian.