SIAM J. COMPUT.
Vol. 13, No. 2, May 1984

010

THE SPECTRA OF FIRST-ORDER SENTENCES AND COMPUTATIONAL COMPLEXITY*

ETIENNE GRANDJEAN†

Abstract. The spectrum of a first-order sentence is the set of cardinalities of its finite models. We refine the well-known equality between the class of spectra and the class of sets (of positive integers) accepted by nondeterministic Turing machines in polynomial time. Let Sp $(d\forall)$ denote the class of spectra of sentences with d universal quantifiers. For any integer $d \geqq 2$ and each set of positive integers, A, we obtain:

$$A \in \text{NTIME}\,(n^d) \to A \in \text{Sp}\,(d\forall) \to A \in \text{NTIME}\,(n^d (\log n)^2).$$

Further the first implication holds even if we use multidimensional nondeterministic Turing machines. These results hold similarly for generalized spectra. As a consequence, we obtain a simplified proof of a hierarchy result of P. Pudlák about (generalized) spectra. We also prove that the set of primes is the spectrum of a certain sentence with only one variable.

Key words. first-order sentences, spectrum, generalized spectrum, computational complexity, nondeterministic Turing machine

Introduction. The spectrum of a first-order sentence is the set of cardinalities of its finite models. If instead of cardinalities of models, we conserve some relations and functions of the models, then we obtain a generalized spectrum. More precisely, let φ be a first-order sentence with relation and function symbols $U_1, \cdots, U_k, V_1, \cdots, V_p$; the generalized spectrum of φ is the set of finite structures $\mathcal{M}$ of type $\{U_1, \cdots, U_k\}$ which have an expansion $\langle \mathcal{M}, V_1, \cdots, V_p \rangle$ satisfying φ.

There are equivalence between certain model-theoretic concepts such as (generalized) spectra and complexity classes. Jones and Selman [11] proved that if A is a set of positive integers,

(a) A is a spectrum iff $A \in \bigcup_d \text{NTIME}\,(n^d)$.

(n represents the input integer.) Similarly, Fagin [5] proved that if G is an isomorphism invariant set of structures of a given type,

(b) G is a generalized spectrum iff $G \in \bigcup_d \text{NTIME}\,(m^d)$.

(m represents the cardinality of the input structure.) More recently, Immerman [10] gave purely logic characterizations of the classes P and PSPACE.

In the present paper we adopt the philosophy expressed by N. Immerman in [9], [10] and J. Lynch in [13]; that is, logical sentences act like automata. Let $\mathcal{P}$ be a property (of integers, of graphs, $\cdots$). According to our choice of a logical or a computational viewpoint, there are two kinds of complexity for $\mathcal{P}$:

(*) the "complexity" of the sentences which characterize property $\mathcal{P}$;

(**) the computational complexity of the automata which recognize property $\mathcal{P}$.

Connections between complexities (*) and (**) allow one to translate some automata-theoretic results into model-theoretic results: Pudlák [15] uses Cook's hierarchy theorem [4] in an essential manner to prove that there is a strict hierarchy of generalized spectra depending on the number of quantifiers. Conversely, there are potential translations of model-theoretic results into computational ones. For example, by equivalence (a), if there is a spectrum whose complement is not a spectrum, then P $\neq$ NP. However, we do not know any proved computational result whose proof uses model-theoretic arguments in an essential manner: we think the reason is that automata and their computations are more "supple" concepts than sentences and their models.

* Received by the editors February 25, 1981, and in revised form December 3, 1982.

† 3 rue de Sévigné, 69003 Lyon, France. Now at Université Claude Bernard–Lyon 1, 69622 Villeurbanne Cedex, France.

The logical concepts that we investigate in this paper are exclusively spectra and generalized spectra which are sets of directed graphs. We justify this last restriction by the fact that results for directed graphs can be easily generalized to other types of structures, and that most natural problems of structures concern (directed) graphs.

We think that there are two natural complexity measures of the (generalized) spectrum of a sentence:

($*_1$) the maximum arity of the relation and function symbols of the sentence;

($*_2$) the number of quantifiers, or, equivalently , of universal quantifiers of the sentence.

Concerning the relationship between complexities ($*_1$) and (**), there is:

THEOREM 0.1 (J. Lynch [13]). *If a set, A, of positive integers belongs to* NTIME (n^d) *for an integer $d \geqq 2$, then A is the spectrum of a sentence with relation symbols of arity at most d and without function symbols. (The same result holds for generalized spectra.)*

This is a nice result because we easily see that any improvement of it would imply an improvement of the inclusion NTIME $(n^d) \subseteq$ DSPACE (n^d). However, we do not know any kind of converse: for example, if a sentence has only binary relation symbols, we do not know any fixed polynomial time upper bound for its spectrum. (See [6] for more details about the hypothetical hierarchy of spectra depending on arity of relation symbols.)

We improve a theorem of Pudlák [15] which connects complexities ($*_2$) and (**) in both directions. Let Sp $(d\forall)$ (resp. GenSp $(d\forall)$) denote the class of spectra (resp. generalized spectra) of sentences with at most d universal quantifiers.

THEOREM 0.2 (Pudlák). *Let G be an isomorphism invariant set of directed graphs. Then, for all integers $d \geqq 2$:*

$$G \in \text{GenSp}\,(d\forall) \textit{ implies } G \in \text{NTIME}\,(m^{3d}),$$

$$G \in \text{NTIME}\,(m^d) \textit{ implies } G \in \text{GenSp}\,(2d\forall).$$

Pudlák also states the following translational lemma of model theory:

LEMMA 0.3 (Pudlák). *For all integers $d \geqq 2$, $e \geqq 1$,*

$$\text{GenSp}\,(d\forall) = \text{GenSp}\,(d+1\forall) \textit{ implies } \text{GenSp}\,(ed\forall) = \text{GenSp}\,(e(d+1)\forall).$$

From these results and from Cook's hierarchy theorem, Pudlák deduces the nice hierarchy result mentioned above which can be reformulated as follows:

COROLLARY 0.4 (Pudlák). *For all integers $d \geqq 0$,*

$$\text{GenSp}\,(d\forall) \subsetneqq \text{GenSp}\,(d+1\forall).$$

Unfortunately, Pudlák's paper [15] does not provide the proofs of the results that he states. Therefore the present paper includes an explicit proof of Pudlák's hierarchy result. However, its main merit is that it considerably improves the connections Pudlák states between GenSp $(d\forall)$ and NTIME (–) since we prove:

THEOREM 0.5. *Let A be a set of positive integers and G be an isomorphism invariant set of directed graphs. Then for all integers $d \geqq 2$:*

(i) $A \in \text{NTIME}\,(n^d) \rightarrow A \in \text{Sp}\,(d\forall) \rightarrow A \in \text{NTIME}\,(n^d (\log n)^2)$;

(ii) $G \in \text{NTIME}\,(m^d) \rightarrow G \in \text{GenSp}\,(d\forall) \rightarrow G \in \text{NTIME}\,(m^d (\log m)^2)$.

These results are interesting for two reasons:

(1) They state that the polynomial degree of the (nondeterministic) time complexity of a property $\mathcal{P}$ is almost equal to the number of universal quantifiers required to express $\mathcal{P}$.

(2) They allow one to prove Pudlák's hierarchy result (and the similar result for spectra) without any model-theoretic lemma, by an immediate translation of the nondeterministic time hierarchy theorem (Cook [4], Seiferas et al. [17]).

Note that the second implication of (i) is essentially proved by H. Lewis [12] in a different context: he investigates the complexity of the satisfiability problem for classes of quantificational sentences.

These results are optimal in the following sense: each first implication of (i) and (ii) holds even if the nondeterministic Turing machine (NTM) is multidimensional; so any improvement of one of the above implications would improve the known simulation of a multidimensional $T(n)$ time-bounded NTM by a (one-dimensional) $T(n) \cdot (\log T(n))^2$ time-bounded NTM.

Notice that Immerman [9], [10] also investigates the number of quantifiers as a complexity measure: using connections between first-order expressibility and computational complexity, he hopes to translate into computational complexity, some lower bounds he obtains for first-order expressibility of "natural" properties of graphs. However, his results and methods are quite different from ours because he characterizes a property, not by only *one* sentence, but by a *uniform sequence* of sentences. (Of course, by the uniformity condition, such a sequence can be regarded in a certain sense as a unique sentence.) As a consequence, he no longer needs additional relation and function symbols, but only a successor relation. Immerman's opinion in [9] is that "it is difficult to show lower bounds for the expressibility of (existential) second-order sentences" and that "first-order sentences mimic computations much more closely." In fact, from our results and from the time hierarchy theorem, it is immediate that there is a spectrum in Sp $(d\forall)$ which cannot be accepted by an NTM in time less than n^d. However, Immerman is right in a certain sense: we are not able to prove a nontrivial lower bound for any naturally defined (generalized) spectrum.

Our paper includes the following sections. Notation and definitions are given in § 1. In § 2, we give two arguments for the naturalness of the measure (Gen)Sp $(d\forall)$: first that it is preserved under intersection and union, secondly that it is equivalent to other complexity measures such as quantifier depth. We also prove the previously mentioned upper bound of spectra.

The announced lower bound for spectra is proved in § 3. Besides the usual "folding" technique [5], [11], [13] for encoding the time units and the tape cells of a computation of a NTM, the proof essentially uses a "numbering" of the ordered pairs $(H(t), t)$, where $H(t)$ is the position of a tape head at instant t; informally, this numbering, N, is such that if $N(x) = (H(t), t)$, and if t' is the first time after t such that $H(t) = H(t')$, then $N(x+1) = (H(t'), t')$.

Lastly, § 4 presents two corollaries of the previous results. First is the hierarchy result of the classes (Gen)Sp $(d\forall)$. The second is a rather surprising result: The set of primes and most "natural" sets of positive integers are spectra of certain sentences with only one variable.

1. Preliminaries.

1.1. Preliminaries in logic. We will use the usual notation and definitions in first-order logic and model theory (see [3, Chap. 1], for example). In particular, our formulas include the equality symbol $=$, and relation and function symbols.

The *arity* of a relation or function symbol is a nonnegative integer; in fact, a 0-ary relation (resp. function) symbol is a proposition (resp. an (individual) constant) symbol.

Our logical connectives are exlusively $\vee$, $\wedge$, $\neg$, interpreted as "or," "and," "not," respectively. The existential and universal quantifiers $\exists$ and $\forall$ are interpreted as "there exists" and "for all," respectively.

(Individual) variables are called x, y, z, t with or without subscripts or primes. The metavariable v (with or without subscripts or primes) will denote any variable.

A *term* is an expression constructed from variables and function symbols in the usual way. An *atomic formula* is of the form $\tau_1 = \tau_2$ or $R(\tau_1, \cdots, \tau_r)$, where τ_i is a term and R is an r-ary relation symbol. A (*first-order*) *formula* is built out of atomic formulas in the usual way, using $\vee$, $\wedge$, $\neg$, $\exists$, $\forall$. A *signed atomic* formula is an atomic formula or its negation.

We suppose familiarity with the notions of subformula, of (existentially or universally) quantified variable, of free occurrence of a variable, and of free variable. A (*first-order*) *sentence* is a formula all of whose variables are quantified. We use $\varphi(v_1, \cdots, v_k)$ to denote a formula φ whose free variables form a subset of $\{v_1, \cdots, v_k\}$.

A *prenex sentence* is a sentence φ of the form

$$Q_1 v_1 \cdots Q_k v_k \psi(v_1, \cdots, v_k)$$

where ψ is a quantifier-free formula and $Q_1, \cdots, Q_k$ are quantifiers; $Q_1 v_1 \cdots Q_k v_k$ and ψ are respectively the *prefix* and the *matrix* of φ. A quantifier-free formula is in *disjunctive normal form* if it is a disjunction of conjunctions of signed atomic formulas.

Sometimes we will use the following abbreviations: $v \neq v'$ for $\neg v = v'$; $\varphi \to \psi$ for $\neg\varphi \vee \psi$; $\varphi \leftrightarrow \psi$ for $(\varphi \to \psi) \wedge (\psi \to \varphi)$. The conjunction of the indexed formulas φ_i, for $i \in J$ and J a finite set, will be denoted $\bigwedge_{i \in J} \varphi_i$, and similarly for the disjunction. Let $\bar{v}_k$ and $Q\bar{v}_k$ (where Q is a quantifier) abbreviate the k-tuple $v_1, \cdots, v_k$ and the string $Qv_1 \cdots Qv_k$, respectively. (More generally, let $\bar{a}_k$ denote a k-tuple of elements $a_1, \cdots, a_k$ in a given set.)

A *type* $\mathcal{T}$ is a finite set of relation and function symbols $\{V_1, \cdots, V_k\}$. The *arity* of $\mathcal{T}$ is the maximum arity of $V_1, \cdots, V_k$. A formula is *of type* $\mathcal{T}$ if all its relation and function symbols are in $\mathcal{T}$.

A *structure* $\mathcal{M} = \langle D, V_1, \cdots, V_k \rangle$ *of type* $\mathcal{T}$ consists of a nonempty set D called the *domain* of $\mathcal{M}$ (denoted $D(\mathcal{M})$) and for each r-ary relation (resp. function) symbol of $\mathcal{T}$, an interpretation, i.e., an r-ary relation (resp. function) on D. If $V_{k+1}, \cdots, V_p$ are other relations and functions on D, the structure $\langle D, V_1, \cdots, V_k, V_{k+1}, \cdots, V_p \rangle$ is called an *expansion* of $\mathcal{M}$ and is denoted $\langle \mathcal{M}, V_{k+1}, \cdots, V_p \rangle$. For convenience, our notation does not distinguish between a relation or function symbol and its interpretation. The cardinality of a structure is the cardinality of its domain. A *finite structure* is a structure of finite cardinality.

Let φ and $\mathcal{M}$ be respectively a sentence and a structure of type $\mathcal{T}$. We put $\mathcal{M} \models \varphi$, and we say that $\mathcal{M}$ is a *model* of φ, to mean that φ becomes a true assertion when each logical symbol $\vee$, $\wedge$, $\neg$, $\exists$, $\forall$, $=$ is given its usual meaning and each relation (resp. function) symbol is given its interpretation in the structure $\mathcal{M}$.

Let D be a nonempty finite set and $\mathcal{T}$ be a type; we will regard the elements of D as constant symbols; let $\mathcal{T}_D = \mathcal{T} \cup D$ be the type $\mathcal{T}$ enlarged by these constant symbols. Any structure of type $\mathcal{T}$ on the domain D is identified with its expansion of type $\mathcal{T}_D$ where constant symbols $a \in D$ are interpreted as themselves. Let $\varphi(\bar{v}_k)$ be a formula of type $\mathcal{T}$ and $\bar{a}_k$ be a k-tuple of elements in D; $\varphi(\bar{a}_k)$ will denote the

sentence of type $\mathcal{T}_D$ constructed from $\varphi(\bar{v}_k)$ be replacing each free occurrence of v_i by a_i. If there is a structure, $\mathcal{M}$, of type $\mathcal{T}$ on the domain D, such that $\mathcal{M} \models \varphi(\bar{a}_k)$, we will say that $\varphi(\bar{a}_k)$ is *satisfiable in* D, or, in case the domain D is implicit, *satisfiable*.

Existential second-order sentences are expressions of the form $\exists V_1 \cdots \exists V_p \varphi$ where φ is a first-order sentence and $V_1, \cdots, V_p$ are among the relation and function symbols of φ. (Unless stated otherwise, formula and sentence will mean first-order formula and first-order sentence.) Let $\mathcal{M}$ be a structure of type $\mathcal{T}$ and ψ be a sentence of type $\mathcal{T} \cup \{V_1, \cdots, V_p\}$, where $V_1, \cdots, V_p$ are symbols not in $\mathcal{T}$. Then we put

$$\mathcal{M} \models \exists V_1 \cdots \exists V_p \psi$$

to mean that $\mathcal{M}$ has an expansion $\mathcal{M}'$ of type $\mathcal{T} \cup \{V_1, \cdots, V_p\}$ such that $\mathcal{M}' \models \psi$. (Of course, the type of $\exists V_1 \cdots \exists V_p \psi$ is $\mathcal{T}$.)

Two (first-order or second-order) sentences of type $\mathcal{T}$, φ and ψ, are (*semantically*) *equivalent* if for each structure $\mathcal{M}$ of type $\mathcal{T}$,

$$\mathcal{M} \models \varphi \quad \text{iff} \quad \mathcal{M} \models \psi.$$

We similarly define the (*semantical*) *equivalence* of formulas $\varphi(\bar{v}_k)$ and $\psi(\bar{v}_k)$.

In the following, for each integer $n > 0$, D_n will denote the set $\{0, 1, \cdots, n-1\}$.

The *spectrum* of a sentence φ, denoted $\mathrm{Sp}(\varphi)$, is the set of cardinalities of its finite models, or, equivalently, the set of integers $n > 0$ such that

$$\langle D_n \rangle \models \exists V_1 \cdots \exists V_p \varphi,$$

if φ is of type $\{V_1, \cdots, V_p\}$.

A *directed graph* or, in short, a *graph*, will be a finite structure $\mathcal{G} = \langle D_m, \mathbf{R} \rangle$, where $\mathbf{R}$ is a binary relation. (In graph-theoretic terminology, $\mathcal{G}$ is a labeled directed graph on m vertices labeled $0, 1, \cdots, m-1$.)

The *generalized spectrum* of a sentence φ of type $\{\mathbf{R}, V_1, \cdots, V_p\}$ where $\mathbf{R}$ is a specified binary relation symbol, is the set of directed graphs $\mathcal{G} = \langle D_m, \mathbf{R} \rangle$ such that

$$\mathcal{G} \models \exists V_1 \cdots \exists V_p \varphi.$$

It is denoted $\mathrm{GenSp}(\varphi)$.

We will investigate the number of universal quantifiers as a complexity measure. There is a little difficulty: a universal (resp. existential) quantifier in the scopes of an odd number of negation signs must be treated as an existential (resp. universal) quantifier. So, to make sense, we always assume in this paper that no quantifier occurs in the scope of a negation sign. (Clearly, each formula is equivalent to a formula of the requisite form.)

Let $\mathrm{Sp}(d\forall)$ (resp. $\mathrm{GenSp}(d\forall)$) denote the class of spectra (resp. generalized spectra) of sentences with at most d universal quantifiers.

The *depth* (resp. *universal depth*) of a formula φ, denoted by $\mathrm{depth}(\varphi)$ (resp. $\forall$-$\mathrm{depth}(\varphi)$), is the maximum number of quantifiers (resp. universal quantifiers) in the scopes of which a subformula of φ can be found. More formally, for a quantifier-free formula ψ, $\mathrm{depth}(\psi) = 0$; for any formulas ψ, ψ_0, ψ_1, $\mathrm{depth}(\exists v \psi) = \mathrm{depth}(\forall v \psi) = 1 + \mathrm{depth}(\psi)$, and $\mathrm{depth}(\psi_0 \vee \psi_1) = \mathrm{depth}(\psi_0 \wedge \psi_1) = \max[\mathrm{depth}(\psi_0), \mathrm{depth}(\psi_1)]$. $\forall$-$\mathrm{depth}(-)$ is defined as $\mathrm{depth}(-)$, except that $\forall$-$\mathrm{depth}(\exists v \psi) = \forall$-$\mathrm{depth}(\psi)$.

1.2. Preliminaries in computational complexity. For all real numbers r, $\lceil r \rceil$ (resp. $\lfloor r \rfloor$) is the least (resp. greatest) integer $n \geqq r$ (resp. $n \leqq r$) and $\log r$ is the logarithm of r in base 2.

Let $f(n)$ and $g(n)$ be two nonnegative real-valued functions on positive integers. We use the notation $f(n) = O(g(n))$ to mean that there is a constant number c such that, for all sufficiently large integers n, $f(n) \leqq cg(n)$.

We suppose that the reader knows the main definitions about Turing machines and nondeterministic computations (see [8, Chap. 7]). Our model of computation will be a one-dimensional nondeterministic Turing machine (NTM); more precisely, an NTM has several one-dimensional tapes, infinite to the right only, which consist of one (read-only) input tape and several (read-write) worktapes. A *multidimensional* NTM is an NTM whose worktapes are multidimensional. (Unless otherwise specified, NTM will mean a one-dimensional NTM.)

Let Σ be a finite set. Σ^+ will denote the set of finite nonempty words over the alphabet Σ. A subset of Σ^+ is called a *language over* Σ.

Let M be an NTM and Σ be the set of input symbols of M. (An input of M is a word of Σ^+.) M *accepts* an input w if, when it is started in start state with all tape cells blank except that the leftmost input tape cells contain w, and with all tape heads at the leftmost cells of the tapes, a computation (i.e., some sequence of moves) takes M to the accepting state. Further, if T is the number of moves of such a computation, then we say that M *accepts* w *in time* T.

Let $T(n)$ be a function from positive integers to positive integers. An NTM M *accepts* a language $L \subseteq \Sigma^+$ *in time* $T(n)$, if

(i) M accepts no input except the words of L, and

(ii) each word of L of length n is accepted by M in time at most $T(n)$.

To make sense, we require that $T(n) \geqq n+1$ since it needs $n+1$ moves to read an input of length n and the first blank symbol. Let NTIME $(f(n))$ denote the class of languages accepted by a NTM in time $\max(n+1, \lceil f(n) \rceil)$, for a real-valued function f.

Let A be a set of positive integers. Identifying each positive integer n with 1^n, (1^n is the word of n 1's), we can regard A as a language over the one-letter alphabet $\Sigma = \{1\}$. Thus A belongs to NTIME $(T(n))$ if the corresponding language belongs to this class. Let Bin (A) denote the set of binary representations of the integers of A. (Of course, Bin $(A) \subseteq \{0, 1\}^+$.)

Our complexity results for spectra will be presented with integers in unary notation. However, if one prefers binary notation, then the following lemma will give an immediate translation of our results.

LEMMA 1.1. *Let A be a set of positive integers. Then for all integers $d \geqq 1$ and $k \geqq 0$, the following statements are equivalent*:

(i) $A \in \text{NTIME}\,(n^d (\log n)^k)$.

(ii) *There is an* NTM *which accepts* Bin (A) *in time* $O(n^d (\log n)^k)$, *where n is the input integer.*

(iii) Bin $(A) \in \text{NTIME}\,(2^{dn} n^k)$, *where n is the length of the input integer.*

Proof. It is sufficient to remark that there is a deterministic Turing machine which transforms any integer n from unary to binary notation (resp. from binary to unary notation) in time $O(n)$: the lemma follows by linear speed-up (see respectively [7], [8] and [2], [17] for linear speed-up of nonlinear and linear functions). □

It is natural to *encode* a directed graph, $\mathcal{G}$, of domain D_m, with the word $w_1 w_2 \cdots w_{m^2}$ over the alphabet $\{0, 1\}$, defined by: for $1 \leqq i \leqq m^2$,

$$w_i = 1 \quad \text{iff} \quad \mathcal{G} \models R(a_1, a_2),$$

where (a_1, a_2) is the ith element of D_m^2 for lexicographical ordering. An NTM, M, *accepts* a set of graphs, G, *in time* $T(m)$ if M accepts in time $T(m)$ the set of words $w_1 \cdots w_{m^2}$ which encode the graphs of G. (Notice that the input has length m^2, not m; m is the number of vertices of the encoded graph.) We will say that a set of graphs, G, belongs to NTIME $(f(m))$ if an NTM accepts G in time $\max(m^2+1, \lceil f(m) \rceil)$.

2. Upper bounds for spectra. Lemmas, Propositions and Theorems 2.1 to 2.5 will be expressed for spectra. However, they hold as well for generalized spectra with the same proofs.

LEMMA 2.1. *For each sentence φ with at most d universal quantifiers, there is a quantifier-free formula $\varphi'(\bar{x}_d)$ such that*

$$\mathrm{Sp}(\varphi) = \mathrm{Sp}(\forall \bar{x}_d \varphi'(\bar{x}_d)).$$

Proof. By standard manipulation of quantifiers, it is easy to put all the quantifiers of φ in front of the sentence. Therefore we can assume that φ is prenex. We use the following fact: a sentence of the form $\forall \bar{v}_k \exists v' \psi(\bar{v}_k, v')$ is equivalent to the second-order sentence $\exists F \forall \bar{v}_k \psi(\bar{v}_k, F(\bar{v}_k))$ where F is a new k-ary function symbol. For example, the sentence $\forall x \exists y \forall z \exists t \psi(x, y, z, t)$ is equivalent to $\exists F_1 \forall x \forall z \exists t \psi(x, F_1(x), z, t)$ and then to

$$\exists F_1 \exists F_2 \forall x \forall z \psi(x, F_1(x), z, F_2(x, z)).$$

From this example, we clearly see that the prenex sentence φ can be transformed according to the following rule: To each existentially quantified variable v, associate a function symbol F (a "Skolem function") and replace each occurrence of v in the matrix of φ by the term $F(v_1, \cdots, v_k)$, where $v_1, \cdots, v_k$ are the universally quantified variables lying before v in the prefix of φ; lastly, remove all the existential quantifiers. □

It is natural to require that complexity classes be closed under intersection and union. We have:

PROPOSITION 2.2. *Let A, B be two sets in* $\mathrm{Sp}(d\forall)$, *for $d \geq 1$, and A' be a finite modification of A (i.e., A' is constructed from A by adding or removing finitely many positive integers). Then*

(i) $A \cap B \in \mathrm{Sp}(d\forall)$;

(ii) $A \cup B \in \mathrm{Sp}(d\forall)$;

(iii) $A' \in \mathrm{Sp}(d\forall)$.

Proof. (i) By Lemma 2.1, we can assume that $A = \mathrm{Sp}(\forall \bar{x}_d \psi_0(\bar{x}_d))$ and $B = \mathrm{Sp}(\forall \bar{x}_d \psi_1(\bar{x}_d))$, where ψ_0 and ψ_1 are quantifier-free formulas. We can also assume that ψ_0 and ψ_1 have no common relation or function symbol. Then $A \cap B = \mathrm{Sp}(\forall \bar{x}_d(\psi_0(\bar{x}_d) \wedge \psi_1(\bar{x}_d)))$. This proves (i).

(ii) Clearly, $A \cup B = \mathrm{Sp}(\varphi)$ with $\varphi = \forall \bar{x}_d \psi_0(\bar{x}_d) \vee \forall \bar{x}_d \psi_1(\bar{x}_d)$. Let R_0 be a new 0-ary relation symbol. (Intuitively, the proposition symbol R_0 stands for the disjunct $\forall \bar{x}_d \psi_0(\bar{x}_d)$.) φ is equivalent to the second-order sentence $\exists R_0[(\neg R_0 \vee \forall \bar{x}_d \psi_0(\bar{x}_d)) \wedge (R_0 \vee \forall \bar{x}_d \psi_1(\bar{x}_d))]$. The first-order sentence in brackets is equivalent to the sentence

$$\varphi' = \forall \bar{x}_d[(\neg R_0 \vee \psi_0(\bar{x}_d)) \wedge (R_0 \vee \psi_1(\bar{x}_d))],$$

and then $A \cup B = \mathrm{Sp}(\varphi')$.

(iii) By (i) and (ii), it is sufficient to prove that for each positive integer k, the sets $\{n : n > k\}$ and $\{k\}$ belong to $\mathrm{Sp}(1\forall)$. We have

$$\{n : n > k\} = \mathrm{Sp}\left(\exists x_0 \cdots \exists x_k \bigwedge_{0 \leq i < j \leq k} x_i \neq x_j\right),$$

$$\{k\} = \mathrm{Sp}\left(\exists x_1 \cdots \exists x_k \left(\bigwedge_{1 \leq i < j \leq k} x_i \neq x_j \wedge \forall y \bigvee_{i=1}^{k} y = x_i\right)\right). \qquad \square$$

Another argument for the naturalness of a complexity measure is the fact that it has several reformulations.

THEOREM 2.3. *Let A be a set of positive integers. $A \in \mathrm{Sp}(d\forall)$ if and only if there is a sentence φ such that $A = \mathrm{Sp}(\varphi)$ with the following property* (i) (*resp.* (ii), (iii)):

(i) $\forall$-depth $(\varphi) = d$;

(ii) *φ has d quantifiers*;

(iii) depth $(\varphi) = d$.

Theorem 2.3 is an immediate consequence of

PROPOSITION 2.4. *For each sentence φ of universal depth d, there is a sentence φ' with d universal quantifiers only and no existential quantifier, so that* $\mathrm{Sp}(\varphi) = \mathrm{Sp}(\varphi')$.

Proposition 2.4 is a particular case of Lemma 2.4′ proved in the Appendix. The proof uses additional "Skolem functions" and relations, as do the proofs of Lemma 2.1 and Proposition 2.2(ii), respectively. Notice that Proposition 2.2 (except part (iii)) is an immediate consequence of Theorem 2.3.

To prove our upper bound theorem for spectra, we shall use the following definitions and lemma.

DEFINITIONS. An *elementary formula of type $\mathcal{T}$* is a signed atomic formula of one of the five following forms:

$$(\neg)v_1 = v_2, \quad (\neg)R(\bar{v}_r), \quad F(\bar{v}_r) = v_{r+1},$$

where R, F are respectively r-ary relation and function symbols of $\mathcal{T}$.

Let n be a strictly positive integer. An *n-formula of type $\mathcal{T}$* is a formula of type $\mathcal{T}_n = \mathcal{T} \cup D_n$ which is constructed from an elementary formula of type $\mathcal{T}$ by replacing each variable by an element of D_n. (So an n-formula of type $\mathcal{T}$ is of the form $(\neg)e_1 = e_2$, $(\neg)R(\bar{e}_r)$ or $F(\bar{e}_r) = e_{r+1}$, where $R, F \in \mathcal{T}$ and $e_i \in D_n$.)

Remark. Since n has length $O(\log n)$ *in base* 2, *an* n-formula can be encoded with $O(\log n)$ symbols.

LEMMA 2.5. *If $A \in \mathrm{Sp}(d\forall)$, then there are a type $\mathcal{T}$ and a sentence*

$$\varphi = \forall \bar{x}_d \exists \bar{y}_k \bigvee_{i=1}^{c} \psi_i(\bar{x}_d, \bar{y}_k)$$

such that:

(i) $A = \mathrm{Sp}(\varphi)$;

(ii) *each ψ_i, $1 \leqq i \leqq c$, is a conjunction of elementary formulas of type $\mathcal{T}$.*

Proof. By Lemma 2.1, $A = \mathrm{Sp}(\forall \bar{x}_d \varphi'(\bar{x}_d))$ for a quantifier-free formula φ'. We construct φ as follows. First put φ' in disjunctive normal form. Secondly, transform the signed atomic subformulas of φ' as in the following example: the subformula $\neg R(F_1(x, y), F_2(y))$ is replaced by the equivalent formula

$$\exists z \exists t(F_1(x, y) = z \wedge F_2(y) = t \wedge \neg R(z, t)).$$

Lastly, put the added existential quantifiers in the prefix. □

THEOREM 2.6. *Let A be a set of positive integers and G be an isomorphism invariant set of directed graphs. Then*:

(i) *$A \in \mathrm{Sp}(d\forall)$ implies $A \in \mathrm{NTIME}(n^d(\log n)^2)$, for each integer $d \geqq 1$;*

(ii) *$G \in \mathrm{GenSp}(d\forall)$ implies $G \in \mathrm{NTIME}(m^d(\log m)^2)$, for each integer $d \geqq 2$.*

Proof. of (i) Let $A = \mathrm{Sp}(\varphi)$ where the sentence φ of type $\mathcal{T}$ is of the form

$$\forall \bar{x}_d \exists \bar{y}_k \bigvee_{i=1}^{c} \psi_i(\bar{x}_d, \bar{y}_k)$$

of Lemma 2.5. The principle of the algorithm which checks if an integer belongs to A, is given by the following equivalences.

Let n be a positive integer and $\mathcal{M}$ be a structure of type $\mathcal{T}$ on the domain D_n. Then:
$\mathcal{M} \models \varphi \leftrightarrow$ for all $\bar{a}_d \in D_n^d$, there are $\bar{b}_k \in D_n^k$ and $i \in \{1, \cdots, c\}$, such that

$$\mathcal{M} \models \psi_i(\bar{a}_d, \bar{b}_k)$$

$\leftrightarrow$ there are a function $g: D_n^d \to D_n^k$ and a function $h: D_n^d \to \{1, \cdots, c\}$, such that

$$\mathcal{M} \models \bigwedge_{\bar{a}_d \in D_n^d} \psi_{h(\bar{a}_d)}(\bar{a}_d, g(\bar{a}_d)).$$

Let $\varphi_{g,h}$ denote the above conjunction. Clearly, $\varphi_{g,h}$ is of the form $\bigwedge_{i=1}^{n'} \pi_i$, where each π_i is an n-formula of type $\mathcal{T}$ and $n' = O(n^d)$. (Here and in the following, the constant numbers implicit in O-notation only depend on the sentence φ.)

We have:

$$n \in \mathrm{Sp}(\varphi) \leftrightarrow \varphi \text{ has a model of domain } D_n,$$

and then from the previous equivalences:

(*) $n \in \mathrm{Sp}(\varphi) \leftrightarrow$ there are functions $g: D_n^d \to D_n^k$ and $h: D_n^d \to \{1, \cdots, c\}$ such that the conjunction $\varphi_{g,h}$ is satisfiable in D_n.

For each (relation or function) symbol $s \in \mathcal{T}$, let Π_s denote the set of n-formulas $\pi_i (1 \leqq i \leqq n')$ mentioned above which contain the symbol s. The following equivalences are obvious:

$$\bigwedge_{s \in \mathcal{T}} \bigwedge_{\pi \in \Pi_s} \pi \text{ is satisfiable} \leftrightarrow \text{for each } s \in \mathcal{T}, \bigwedge_{\pi \in \Pi_s} \pi \text{ is satisfiable}$$

$\leftrightarrow$ for each $s \in \mathcal{T}$, Π_s includes no pair of incompatible n-formulas.

(Incompatible n-formulas are of the forms $R(\bar{e}_r)$ and $\neg R(\bar{e}_r)$ or $F(\bar{e}_r) = e$ and $F(\bar{e}_r) = e'$ with $e \neq e'$.)

The following nondeterministic algorithm (divided in two procedures, (a) and (b)), thus emerges.

(a) Construct the binary representation of n. Guess (nondeterministically) the functions g and h and write the conjunction $\varphi_{g,h}$. There are n^d values to guess for each function. Since integers are written in base 2, procedure (a) requires a time $O(n^d \log n)$.

(b) Check (deterministically) if $\varphi_{g,h}$ is satisfiable in D_n. (Recall: $\varphi_{g,h} = \bigwedge_{i=1}^{n'} \pi_i$, where $n' = O(n^d)$ and each π_i is an n-formula of type $\mathcal{T}$.) Procedure (b) divides into three steps:

(b_1) Evaluate the (in)equalities, i.e., n-formulas of the form $(\neg) e = e'$, among the n-formulas π_i; if any is false, then $\varphi_{g,h}$ is not satisfiable; otherwise, delete the (true) (in)equalities and sort the conjunction $\varphi_{g,h}$ in the form $\bigwedge_{s \in \mathcal{T}} \bigwedge_{\pi \in \Pi_s} \pi$.

(b_2) Sort each conjunction $\bigwedge_{\pi \in \Pi_s} \pi$ according to the lexicographical order of the arguments $\bar{e}_r$ of the n-formulas π. (Recall that π has form $(\neg) R(\bar{e}_r)$ or $F(\bar{e}_r) = e_{r+1}$.)

(b_3) For each $s \in \mathcal{T}$, test whether Π_s includes a pair of incompatible n-formulas, using the fact that incompatible n-formulas (if any) now appear side by side.

We clearly see that steps (b_1) and (b_3) each require time $O(n^d \log n)$. Therefore procedure (b) requires time $O(n^d (\log n)^2)$ which is the time to execute step (b_2) with the sorting algorithm of [1, p. 78].

It is easy to implement procedures (a) and (b) on an NTM, and then $A \in$ NTIME $(n^d (\log n)^2)$, by linear speed-up.

Proof of (ii) Similar to the proof of (i). Therefore we only emphasize the differences. Let $G = \mathrm{GenSp}(\varphi)$ where the sentence φ of type $\mathscr{T} = \{\mathbf{R}, V_1, \cdots, V_p\}$ is as in Lemma 2.5.

Let $\mathscr{G}$ be a directed graph of domain D_m. By the argument used in the proof of (i), we obtain:

$\mathscr{G} \in \mathrm{GenSp}(\varphi) \leftrightarrow$ there are functions $g: D_m^d \to D_m^k$ and $h: D_m^d \to \{1, \cdots, c\}$ such that $\mathscr{G}$ has an expansion $\langle \mathscr{G}, V_1, \cdots, V_p \rangle$ which is a model of $\varphi_{g,h}$.

Let $\Delta(\mathscr{G})$ denote the conjunction of the m^2 m-formulas $(\neg)\mathbf{R}(e_1, e_2)$, where $(e_1, e_2) \in D_m^2$, such that $\mathscr{G} \models (\neg)\mathbf{R}(e_1, e_2)$. Then we obtain the following equivalence:

(**) $\mathscr{G} \in \mathrm{GenSp}(\varphi) \leftrightarrow$ there are functions $g: D_m^d \to D_m^k$ and $h: D_m^d \to \{1, \cdots, c\}$, such that the conjunction $\varphi_{g,h} \wedge \Delta(\mathscr{G})$ is satisfiable in D_m.

The second member of equivalence (**) is similar to the second member of equivalence (*) in the proof of (i), except that $\varphi_{g,h}$ is now replaced by $\varphi_{g,h} \wedge \Delta(\mathscr{G})$ which is a conjunction of $O(m^d)$ m-formulas of type $\mathscr{T}$, since $d \geqq 2$. Therefore the remainder of the proof is exactly like that of (i). □

Remarks. Part (i) of Theorem 2.6 is essentially stated by H. Lewis [12, Prop. 3.2] with a proof more informal than ours. More precisely, he states the following: "Whether a prenex sentence with d universal quantifiers has a model of cardinality n can be ascertained nondeterministically in time $f(|\varphi| \cdot n^d)$, for some polynomial f." ($|\varphi|$ denotes the length of sentence φ.) However, H. Lewis states his proposition in a different context: he uses it as a tool to prove a complexity upper bound for the satisfiability problem of a class of sentences with a fixed number of universal quantifiers; he does not need to know a precise value of polynomial f. (Moreover he assumes that φ contain no function symbol and no equality symbol.)

By Theorems 2.3 and 2.6, the spectrum of any sentence φ of universal depth d belongs to $\mathrm{NTIME}(n^d (\log n)^2)$. In fact, there is a natural generalization of the algorithm of Theorem 2.6 which accepts $\mathrm{Sp}(\varphi)$ in time $O(n^d (\log n)^2)$, and similarly for generalized spectra.

3. Lower bounds for spectra. We want to "simulate" a computation of an NTM in a finite structure. Therefore we need a numbering of the structure to express the numbering of the tape cells and of the time units the computation requires. We can construct a linear order by:

LEMMA 3.1. *There is a first-order sentence ψ such that*:

(i) *ψ has only two universal quantifiers*;

(ii) *ψ is of type $\mathscr{T} = \{F_{\mathrm{Suc}}, R_<, R_{\mathrm{Suc}}, c_0, c_l\}$, where F_{Suc} is a unary function symbol, $R_<$ and R_{Suc} are binary relation symbols and c_0, c_l are constant symbols*;

(iii) *if $\mathscr{M} = \langle D, R_<, R_{\mathrm{Suc}}, c_0, c_l \rangle$ is a finite structure of cardinality at least two, then*:

$\mathscr{M} \models \exists F_{\mathrm{Suc}} \psi \leftrightarrow$ *$R_<$ is a strict linear order of D, and R_{Suc}, c_0, c_l are respectively the corresponding successor relation and the first and last elements of this order.*

Proof. Let us give some sentences with their intuitive meaning.

ψ_1: $\forall x \exists y F_{\mathrm{Suc}}(y) = x \wedge \forall x F_{\mathrm{Suc}}(x) \neq x$.

ψ_1 expresses that F_{Suc} is a permutation of the (finite) domain and has no loop.

ψ_2: $\forall x \neg R_<(x, x) \wedge \forall x \forall y [x \neq y \to (R_<(x, y) \leftrightarrow \neg R_<(x, y))]$.

ψ_2 expresses that $R_<$ is a tournament.

$$\psi_3: \forall x \forall y[(R_<(x, y) \wedge y \neq c_l) \rightarrow R_<(x, F_{\mathrm{Suc}}(y))] \wedge F_{\mathrm{Suc}}(c_l) = c_0.$$

ψ_3 expresses that $R_<$ is a transitive relation for function F_{Suc}, except for the value $F_{\mathrm{Suc}}(c_l)$ which is c_0.

A consequence of the conjunction $\psi_1 \wedge \psi_2 \wedge \psi_3$ is that the permutation F_{Suc} has only one cycle. In investigating this cycle, we clearly see that this conjunction implies that $R_<$ is a linear order of the domain, with first and last elements c_0, c_l, respectively, and that F_{Suc} maps each element to its immediate successor for this order, with moreover $F_{\mathrm{Suc}}(c_l) = c_0$. Therefore the following sentence defines the successor relation:

$$\psi_4: \forall x \forall y[R_{\mathrm{Suc}}(x, y) \leftrightarrow (F_{\mathrm{Suc}}(x) = y \wedge x \neq c_l)].$$

So the conjunction $\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$ has properties (ii) and (iii). Clearly, it has an equivalent form with property (i) also since it is a conjunction of sentences with at most two universal quantifiers. □

In the following, we will use Lemma 3.1 with the expressive symbols $<$, Suc, 0, l instead of $R_<$, R_{Suc}, c_0, c_l, respectively.

The following is our second main result.

THEOREM 3.2. *Let A be a set of positive integers and G be an isomorphism invariant set of directed graphs. Then, for any integer $d \geqq 2$:*

(i) $A \in \mathrm{NTIME}(n^d)$ *implies* $A \in \mathrm{Sp}(d\forall)$;

(ii) $G \in \mathrm{NTIME}(m^d)$ *implies* $G \in \mathrm{GenSp}(d\forall)$.

Moreover these two implications hold even if we use multidimensional NTMs.

Proof of (i). Let A be a set in NTIME (n^d). By Proposition 2.2(iii), we can assume that A is a set of integers $n \geqq 2$. For each integer $n \geqq 2$, let us consider $\mathfrak{c}1^{n-2}\$$, a word of length n, that we regard as a "self-bounded" unary representation of n. Clearly, the set $A' = \{\mathfrak{c}1^{n-2}\$: n \in A\}$ also belongs to NTIME (n^d), and then, by speed-up, there is an NTM, M, which accepts A' in time $n^d - 1$. The input head of M does not visit any cell outside the input, because of the "bounds" $\mathfrak{c}$ and \$. (An NTM which accepts A must visit the n cells of the input plus the next one to the right. It is less convenient to encode $n+1$ cells than n cells in a domain of n elements: this is why we consider A' in place of A. Similarly, we choose the time bound $n^d - 1$ for technical reasons which will be explained later.)

In the following, we will adopt almost the same notation as J. Lynch [13] used in the proof of his theorem (mentioned in our introduction), so that it is easy to compare his proof and ours.

Suppose that the tapes of M are the input tape and only one (one-dimensional) worktape. The set of input symbols of M is of course $\Sigma = \{1, \mathfrak{c}, \$\}$. Let Γ be the set of worktape symbols of M, including the blank symbol **b**. Let Q be the set of M's states, including q_0, the start state, and q_a, the accepting state. Let δ be the transition function of the NTM M. More precisely, δ maps each $(\sigma, \gamma, q) \in \Sigma \times \Gamma \times Q$ to a subset of $\Gamma \times \{-1, 0, 1\}^2 \times Q$; the set $\delta(\sigma, \gamma, q)$ consists of those $(\gamma', \alpha, \beta, q')$ such that if at some time, the symbols under the input head and the worktape head are respectively σ, γ and M is in state q, then the following is a possible move: M prints γ' in place of γ, and at the following time, the input head and the worktape head move accordingly to numbers α, β, respectively (0 means no movement, 1 and -1 a movement to the right and to the left, respectively) and M enters state q'.

Since M accepts A' in time $n^d - 1$, we can regard each computation of M (on an input of length n) as a sequence of exactly $n^d - 1$ moves by adopting the following conventions: each computation of M is truncated by an $n^d - 1$ time-bounded clock;

in case M enters an (accepting or rejecting) final state before the clock rings, then M continues "running" in the same configuration until the clock rings. Therefore each computation of M uses exactly n^d configurations (including the start and the final configurations) and at most n^d worktape cells.

By Lemma 3.1, there is an existential second-order sentence which defines a linear order $<$ of the domain with the corresponding successor relation, Suc, and first and last elements, 0 and l, respectively. To encode the moves of M, we shall use the lexicographical order of k-tuples, constructed from $<$, and also denoted $<$, and the corresponding successor relation, also denoted Suc. More precisely, $\bar{x}_k < \bar{y}_k$ is an abbreviation of

$$x_1 < y_1 \vee \bigvee_{j=2}^{k} \left(\bigwedge_{i=1}^{j-1} x_i = y_i \wedge x_j < y_j \right).$$

Similarly, $\mathrm{Suc}(\bar{x}_k, \bar{y}_k)$ abbreviates the following formula:

$$\left[\mathrm{Suc}(x_1, y_1) \wedge \bigwedge_{i=2}^{k} (x_i = l \wedge y_i = 0) \right]$$
$$\vee \bigvee_{j=2}^{k-1} \left[\bigwedge_{i=1}^{j-1} x_i = y_i \wedge \mathrm{Suc}(x_j, y_j) \wedge \bigwedge_{i=j+1}^{k} (x_i = l \wedge y_i = 0) \right]$$
$$\vee \left[\bigwedge_{i=1}^{k-1} x_i = y_i \wedge \mathrm{Suc}(x_k, y_k) \right].$$

Clearly $\bar{x}_k < \bar{y}_k$ and $\mathrm{Suc}(\bar{x}_k, \bar{y}_k)$ express the required relations. Let $\bar{y}_k = \bar{x}_k + 1$ and $\bar{x}_k = \bar{y}_k - 1$ be synonymous with $\mathrm{Suc}(\bar{x}_k, \bar{y}_k)$. Lastly, let $\bar{x}_k = \bar{y}_k$ (or $\bar{x}_k = \bar{y}_k + 0$) and $\bar{x}_k \neq \bar{y}_k$ abbreviate the conjunction $\bigwedge_{i=1}^{k} x_i = y_i$ and its negation, respectively.

Let $\bar{v}$ in short denote the d-tuple of variables $\bar{v}_d = v_1, \cdots, v_d$. Similarly, let $\bar{0}$ (resp. $\bar{l}$) denote the constant symbol 0 (resp. l) repeated d times. For convenience, our notation does not distinguish between a variable and its assignment.

Now let us consider a domain of n elements, denoted E_n, and let us intuitively describe how to encode in E_n a computation of M on an input of length n. There are as many elements in E_n^d as time units (resp. worktape cells) used in the computation. So each element $\bar{t}$ (resp. y, $\bar{y}$) of E_n^d (resp. E_n, E_n^d) corresponds to a time unit (resp. an input cell, a worktape cell), also denoted $\tilde{t}$ (resp. y, $\bar{y}$). (This is the "folding" technique of [5], [11], [13].) For convenience, we will use (d, k)-ary functions: a (d, k)-ary function symbol F abbreviates a k-tuple $F_1, \cdots, F_k$ of d-ary function symbols. Let us introduce the following relation and function symbols for which the argument $\bar{t}$ intuitively means "at time $\bar{t}$":

The d-ary relation symbols $C^*_\sigma, C_\gamma, C'_\gamma, \sigma \in \Sigma, \gamma \in \Gamma$: $C^*_\sigma(\bar{t})$ (intuitively) means "the symbol under the input head is σ"; $C_\gamma(\bar{t})$ (resp. $C'_\gamma(\bar{t})$) means "the symbol under (resp. printed by) the worktape head is γ."

The d-ary function symbol H^*: $H^*(\bar{t})$ is "the cell, y, under the input head."

The (d, d)*-ary function symbol* H: $H(\bar{t}) = (H_1(\bar{t}), \cdots, H_d(\bar{t}))$ is "the cell, $\bar{y}$, under the worktape head."

The d-ary relation symbols $S_q, q \in Q$: $S_q(\bar{t})$ means "M is in state q."

For technical reasons, we also introduce a $(d, 2d)$-ary function symbol, N, less intuitive than the previous symbols. It will be used to lexicographically number the n^d couples $(H(\bar{t}), \bar{t})$ of the computation. The (intuitive) value of $N(\bar{x}) = (N_1(\bar{x}), \cdots, N_{2d}(\bar{x}))$ is "the couple $(H(\bar{t}), \bar{t})$ of number $\bar{x}$."

Function N is defined by the two following sentences with only d universal quantifiers:

$$\varphi_1: \forall \bar{t} \exists \bar{x} (H(\bar{t}), \bar{t}) = N(\bar{x}),$$

where the subformula $(H(\bar{t}), \bar{t}) = N(\bar{x})$ abbreviates

$$\bigwedge_{i=1}^{d} H_i(\bar{t}) = N_i(\bar{x}) \wedge \bigwedge_{i=1}^{d} t_i = N_{d+i}(\bar{x});$$

$$\varphi_2: (\forall \bar{x} \neq \bar{l})(\exists \bar{x}' = \bar{x} + 1) N(\bar{x}) < N(\bar{x}'),$$

where $(\forall \bar{x} \neq \bar{l})$ and $(\exists \bar{x}' = \bar{x} + 1)$ abbreviate $\forall \bar{x} (\bar{x} \neq \bar{l} \rightarrow \cdots)$ and $\exists \bar{x}' (\bar{x}' = \bar{x} + 1 \wedge \cdots)$, respectively. Clearly, φ_1 expresses that the equality $(H(\bar{t}), \bar{t}) = N(\bar{x})$ defines a bijection between the n^d d-tuples $\bar{t}$ and the n^d d-tuples $\bar{x}$, and φ_2 expresses that N is an increasing function for lexicographical order.

Figure 1 illustrates (for a particular computation) how function N numbers the ordered pairs $(H(\bar{t}), \bar{t})$ which are intuitively the worktape head positions during the computation. (We assume that $d = 1$ and $n = 6$; each element of E_6 is denoted in Fig. 1 by its rank for order $<$.)

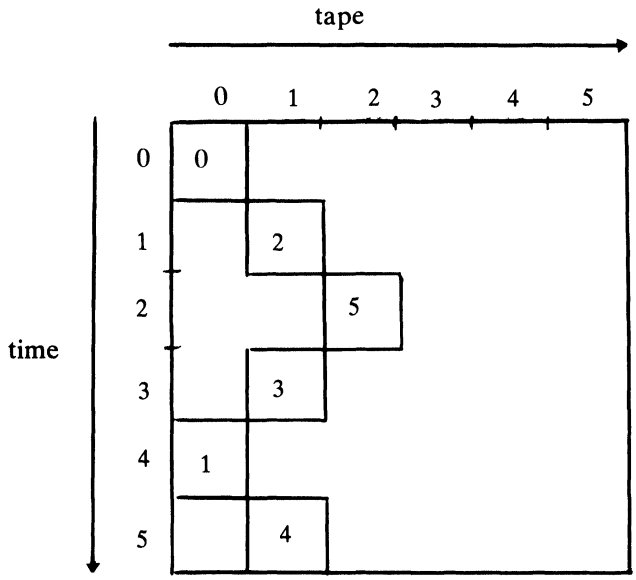

FIG. 1

It should be clear that $\varphi_1 \wedge \varphi_2$ implies that function N satisfies the two following (informal) statements:

(a) $\bar{t}'$ is the first time after $\bar{t}$ such that $H(\bar{t}) = H(\bar{t}')$ iff $H(\bar{t}) = H(\bar{t}')$ and there is $\bar{x}$ such that $N(\bar{x}) = (H(\bar{t}), \bar{t})$ and $N(\bar{x}+1) = (H(\bar{t}'), \bar{t}')$.

(b) $\bar{t}' \neq \bar{0}$ and the worktape cell $H(\bar{t}')$ has never been visited before time $\bar{t}'$ iff there are $\bar{x}, \bar{t}$ such that $N(\bar{x}) = (H(\bar{t}), \bar{t})$ and $N(\bar{x}+1) = (H(\bar{t}'), \bar{t})$ and $H(\bar{t}) \neq H(\bar{t}')$.

The following sentences express how to encode an accepting computation of M in a structure with the relation and function symbols mentioned above.

$$\varphi_3\colon \forall \bar{t}[(H^*(\bar{t}) = 0 \leftrightarrow C^*_{\not\in}(\bar{t})) \wedge (H^*(\bar{t}) = l \leftrightarrow C^*_{\$}(\bar{t})) \wedge (0 < H^*(\bar{t}) < l \leftrightarrow C^*_1(\bar{t}))].$$

φ_3 expresses what is the symbol under the input head. (Recall that the input is $\not\in 1^{n-2}\$$.)

$$\varphi_4\colon \underline{C}_b(\bar{0}) \wedge H^*(\bar{0}) = 0 \wedge H(\bar{0}) = \bar{0} \wedge \underline{S}_{q_0}(\bar{0}).$$

(In φ_4 and in the following sentences, $\underline{C}_\gamma(\bar{t})$, $\gamma \in \Gamma$, abbreviates the conjunction $C_\gamma(\bar{t}) \wedge \bigwedge_{\gamma' \in \Gamma - \{\gamma\}} \neg C_{\gamma'}(\bar{t})$; $\underline{C}'_\gamma(\bar{t})$, $\gamma \in \Gamma$, and $\underline{S}_q(\bar{t}), q \in Q$, are similar abbreviations.) φ_4 describes the start configuration.

Let φ_5 be the conjunction, for all $(\sigma, \gamma, q) \in \Sigma \times \Gamma \times Q$, of:

$$(\forall \bar{t} \neq \bar{l})(\exists \bar{t}' = \bar{t} + 1)$$
$$[(C^*_\sigma(\bar{t}) \wedge C_\gamma(\bar{t}) \wedge S_q(\bar{t}))$$
$$\rightarrow \bigvee (\underline{C}'_{\gamma'}(\bar{t}) \wedge H^*(\bar{t}') = H^*(\bar{t}) + \alpha \wedge H(\bar{t}') = H(\bar{t}) + \beta \wedge \underline{S}_{q'}(\bar{t}'))],$$

where the disjunction extends over all $(\gamma', \alpha, \beta, q')$ of $\delta(\sigma, \gamma, q)$. φ_5 describes the set of possible moves determined by the transition function δ. (Notice that we assume that the worktape head prints the symbol γ' at time $\bar{t}$, not at time $\bar{t} + 1$.)

$$\varphi_6\colon (\forall \bar{t}' \neq \bar{0})\exists \bar{t} \exists \bar{x}(\exists \bar{x}' = \bar{x} + 1)$$
$$[N(\bar{x}) = (H(\bar{t}), \bar{t}) \wedge N(\bar{x}') = (H(\bar{t}'), \bar{t}')$$
$$\wedge \bigwedge_{\gamma \in \Gamma} [(H(\bar{t}) = H(\bar{t}') \wedge C'_\gamma(\bar{t})) \rightarrow \underline{C}_\gamma(\bar{t}')]$$
$$\wedge [H(\bar{t}) \neq H(\bar{t}') \rightarrow \underline{C}_b(\bar{t}')]].$$

Using equivalences (a) and (b), φ_6 expresses what is the symbol under the worktape head at time $\bar{t}' \neq \bar{0}$, according to whether the scanned worktape cell has been visited or not before.

$$\varphi_7\colon S_{q_a}(\bar{l}).$$

Let φ_0 be the sentence ψ of Lemma 3.1 and φ be the conjunction $\bigwedge_{i=0}^{7} \varphi_i$. Then it should be clear that M accepts $\not\in 1^{n-2}\$$ iff φ has a model of cardinality n. Hence $n \in A$ iff $n \in \mathrm{Sp}(\varphi)$. Moreover, φ is a conjunction of sentences with at most d universal quantifiers. This proves part (i) of the theorem in case the machine M has only one (one-dimensional) worktape.

In the general case, for every k-dimensional worktape of M, we need a (d, kd)-ary function symbol, H, and a $(d, (k+1)d)$-ary function symbol, N, and the corresponding sentences φ_1, φ_2 and φ_6. There are also obvious modifications of sentences φ_4 and φ_5.

Proof of (ii). Similar to the proof of (i). Therefore we only dwell on the differences. Let G be an isomorphism invariant set of directed graphs such that $G \in \mathrm{NTIME}(m^d)$. Let Σ be the alphabet of six symbols $\{0, 1, (0, \not\in), (1, \not\in), (0, \$), (1, \$)\}$. A "self-bounded" encoding of a directed graph $\mathcal{G}$ is a word of Σ^{m^2},

$$(w_1, \not\in) w_2 \cdots w_{m^2-1} (w_{m^2}, \$),$$

where the word $w_1 \cdots w_{m^2}$ of $\{0, 1\}^{m^2}$ encodes the graph $\mathcal{G}$. Clearly, there is an NTM which accepts the set of self-bounded encodings of the graphs of G in time $\max(m^2, m^d - 1)$.

We construct a sentence φ such that $G = \mathrm{GenSp}(\varphi)$ exactly as in the proof of (i), except that the conjunct φ_3, expressing what is the symbol under the input head, is now replaced by the conjunction of:

$$\forall \bar{t}[H^*(\bar{t}) = (0,0) \leftrightarrow C^*_{\text{¢}}(\bar{t})],$$

$$\forall \bar{t}[H^*(\bar{t}) = (l,l) \leftrightarrow C^*_{\$}(\bar{t})],$$

$$\forall \bar{t} \exists x \exists y[H^*(\bar{t}) = (x,y) \wedge (\mathbf{R}(x,y) \leftrightarrow C^*_1(\bar{t})) \wedge (\neg \mathbf{R}(x,y) \leftrightarrow C^*_0(\bar{t}))].$$

Note that H^* is now a $(d, 2)$-ary function symbol since we have to "fold up" an input of length m^2 to encode it in a domain of m elements.

There is a slight technical difficulty in case $G \in \mathrm{NTIME}(m^2)$, because an NTM which accepts the set of self-bounded encodings of graphs of G in time m^2 cannot be sped up, and then each computation requires m^2+1 configurations. In this case, the ordered pair $\bar{l} = (l, l)$ does not encode the last time unit, but rather the time unit before last. However, we are not interested by the last configuration, except for its state. From these remarks, the reader should be able to modify sentences φ_5 and φ_7 for this case. □

Remarks. In fact, we have proved a little more than the stated theorem: if A, a set of positive integers, belongs to NTIME (n^d), for an integer $d \geqq 2$, then A is the spectrum of a sentence with at most d universal quantifiers and a type of arity d. Similarly, for generalized spectra.

As P. Pudlák [16] points out, the implication $G \in \mathrm{NTIME}(m^d) \to G \in \mathrm{GenSp}(d\forall)$ might be useful for finding some nontrivial lower bound for a concrete problem in NP. In particular, if $G \notin \mathrm{GenSp}(2\forall)$ then $G \notin \mathrm{NTIME}(m^2)$.

4. Corollaries.

COROLLARY 4.1 (Pudlák [15] for (ii)). *Let d be a nonnegative integer. Then*:

(i) *there is a set of positive integers, A, such that*:

$$A \in \mathrm{Sp}(d+1\forall) - \mathrm{Sp}(d\forall);$$

(ii) *there is a set of directed graphs, G, such that*

$$G \in \mathrm{GenSp}(d+1\forall) - \mathrm{GenSp}(d\forall).$$

Proof. (i) Clearly, $\{1\} \in \mathrm{Sp}(1\forall) - \mathrm{Sp}(0\forall)$. So assume that $d \geqq 1$. A particular case of the nondeterministic time hierarchy [4], [17] is that there is a set of positive integers

$$A \in \mathrm{NTIME}(n^{d+1}) - \mathrm{NTIME}(n^d(\log n)^2).$$

From Theorems 2.6 and 3.2, it immediately follows that

$$A \in \mathrm{Sp}(d+1\forall) - \mathrm{Sp}(d\forall).$$

(ii) Let us consider a sentence φ with $d+1$ universal quantifiers only, such that $A = \mathrm{Sp}(\varphi)$. Let $\mathcal{T}$ be the type of φ. Now let us regard φ as a sentence of type $\mathcal{T} \cup \{\mathbf{R}\}$, $\mathbf{R}$ a new binary relation symbol. Let us define $G(A) = \mathrm{GenSp}(\varphi)$. $G(A)$ is the set of directed graphs on n vertices, such that $n \in A$. Let us assume that $G(A) = \mathrm{GenSp}(\varphi')$ where φ' has at most d universal quantifiers; then $A = \mathrm{Sp}(\varphi')$, a contradiction. □

Remarks. By the same proof, Corollary 4.1 holds not only for directed graphs, but also for structures of any type.

Corollary 4.1 can be strengthened as follows: The sentence φ (with $d+1$ universal quantifiers) such that $A = \mathrm{Sp}(\varphi)$ (resp. $G(A) = \mathrm{GenSp}(\varphi)$), has a type of arity $d+1$.

In analyzing the proof of Theorem 3.2(i), we clearly see that the only reason why we assume $d \geqq 2$ is that we need two universal quantifiers to define a linear order. If we suppose that a structure has a "built-in" linear ordering, then the restriction $d \geqq 2$ can be removed.

DEFINITION. Let φ be a sentence whose type includes $<$, a binary relation symbol. Let $\mathrm{Sp}_<(\varphi)$ denote the set of integers n, such that there is a model of φ of domain $D_n = \{0, \cdots, n-1\}$, where $<$ is interpreted as the natural order of D_n.

LEMMA 4.2. *There is a first-order sentence ψ such that*:

(i) *ψ has only one universal quantifier*;

(ii) *ψ is of type $\mathcal{T} = \{<, F_{\mathrm{Suc}}, c_0, c_l\}$ where $<$ is a binary relation symbol, F_{Suc} is a unary function symbol and c_0, c_l are constant symbols*;

(iii) *Let $\mathcal{M}$ be a structure of type $\mathcal{T}$ on the domain D_n where $<$ is the natural order of D_n. Then*

$$\mathcal{M} \models \psi \text{ iff } c_0 = 0, c_l = n-1 \text{ and } F_{\mathrm{Suc}}(e) = e+1 \text{ for each } e \in D_n,$$
$$\text{except that } F_{\mathrm{Suc}}(n-1) = 0.$$

Proof. Properties (ii) and (iii) clearly hold for the conjunction of

$$\forall x\, \exists y\, x = F_{\mathrm{Suc}}(y) \quad \text{and} \quad (\forall x \neq c_l)x < F_{\mathrm{Suc}}(x) \wedge F_{\mathrm{Suc}}(c_l) = c_0. \qquad \square$$

COROLLARY 4.3. *Let A be a set of positive integers in* NTIME (n). *Then there are a type $\mathcal{T}$ and a quantifier-free formula $\varphi(x)$ of type $\mathcal{T} \cup \{<\}$, such that*:

(i) *φ has only one variable x*;

(ii) $A = \mathrm{Sp}_<(\forall x \varphi(x))$;

(iii) *the arity of $\mathcal{T}$ is* 1.

Proof. Let φ' be the conjunction of the sentence ψ of Lemma 4.2 and of the sentences $\varphi_1, \cdots, \varphi_7$ (of Theorem 3.2) in which each occurrence of $\mathrm{Suc}(v, v')$ is replaced by $F_{\mathrm{Suc}}(v) = v' \wedge v \neq c_l$. (Moreover φ_5 and φ_7 are modified as in case $G \in$ NTIME (m^2) of Theorem 3.2(ii).) Let φ'' be the equivalent form of φ' with only one universal quantifier. $\forall x \varphi(x)$ is the sentence constructed from φ'' as in Lemma 2.1. $\square$

Let Prime denote the set of prime numbers. The usual algorithm for testing whether an integer n (written in base 2, for example) is prime, is to divide n by the $\lfloor \sqrt{n} \rfloor$ first positive integers. This algorithm works in time $\sqrt{n} f(\log n)$, for a polynomial f. (Pratt's nondeterministic algorithm [8, p. 342], [14] works in time $f(\log n)$, for a polynomial f.) From Lemma 1.1, it follows that Prime $\in$ NTIME (n). Indeed it seems that most "natural" sets of integers belong to NTIME (n).

COROLLARY 4.4. *There is a type $\mathcal{T}$ and a prenex sentence φ of type $\mathcal{T} \cup \{<\}$ such that*:

(i) $\mathrm{Prime} = \mathrm{Sp}_<(\varphi)$;

(ii) *φ has only one variable (universally quantified)*;

(iii) *the arity of $\mathcal{T}$ is* 1.

Appendix. We want to prove:

PROPOSITION 2.4. *For each sentence φ of universal depth d, there is a sentence φ' with d universal quantifiers only and no existential quantifier, so that* $\mathrm{Sp}(\varphi) = \mathrm{Sp}(\varphi')$.

LEMMA 2.4′. *For each formula $\varphi(\bar{x}_k)$ of type $\mathcal{T}$, such that $\forall$-depth $(\varphi) = d$, there is a quantifier-free formula $\varphi^*(\bar{x}_k, \bar{y}_d)$ of type $\mathcal{T}^* \supseteq \mathcal{T}$, so that* (i) *and* (ii) *are true*:

(i) *Any finite structure, $\mathcal{M}$, of type $\mathcal{T}$, has an expansion $\langle \mathcal{M}, \mathcal{N} \rangle$ of type $\mathcal{T}^*$, such that for each k-tuple $\bar{a}_k$ in $D(\mathcal{M})$,*

$$\mathcal{M} \models \varphi(\bar{a}_k) \quad \text{implies} \quad \langle \mathcal{M}, \mathcal{N} \rangle \models \forall \bar{y}_d \varphi^*(\bar{a}_k, \bar{y}_d).$$

(ii) *Given any finite structures,* $\mathcal{M}$ *and* $\langle \mathcal{M}, \mathcal{N} \rangle$, *of respective types* $\mathcal{T}$ *and* $\mathcal{T}^*$, *and any* k*-tuple* $\bar{a}_k$ *in* $D(\mathcal{M})$,

$$\langle \mathcal{M}, \mathcal{N} \rangle \models \forall \bar{y}_d \varphi^*(\bar{a}_k, \bar{y}_d) \quad \textit{implies} \quad \mathcal{M} \models \varphi(\bar{a}_k).$$

From Lemma 2.4′, we obtain Proposition 2.4 for each sentence φ, by taking $\varphi' = \forall \bar{y}_d \varphi^*(\bar{y}_d)$.

Proof of Lemma 2.4′. By induction on the "complexity" of the formula φ. We build φ^* as follows:

Case 1. For a quantifier-free formula ψ, $\psi^* = \psi$.

For all formulas $\psi_0(\bar{x}_k)$, $\psi_1(\bar{x}_k)$, $\psi(\bar{x}_k, z)$:

Case 2. $(\psi_0 \wedge \psi_1)^* = \psi_0^* \wedge \psi_1^*$;

Case 3. $(\forall z \psi)^* = \psi^*$;

Case 4. $(\psi_0 \vee \psi_1)^* = [R(\bar{x}_k) \vee \psi_0^*(\bar{x}_k, \bar{y}_d)] \wedge [\neg R(\bar{x}_k) \vee \psi_1^*(\bar{x}_k, \bar{y}_d)]$, where R is a new k-ary relation symbol and $d = \forall$-depth $(\psi_0 \vee \psi_1)$;

Case 5. $(\exists z \psi(\bar{x}_k, z))^* = \psi^*(\bar{x}_k, F(\bar{x}_k), \bar{y}_d)$, where F is a new k-ary function symbol and $d = \forall$-depth (ψ).

We shall prove (i) and (ii) only for the hardest case, Case 4, and shall give a sketch of proof of (i) in Case 5. The reader can easily complete the proof for the other cases.

Case 4. $\varphi = \psi_0 \vee \psi_1$. Let $\mathcal{M}$ be a finite structure of type $\mathcal{T}$ (the type of φ, and also of ψ_0 and ψ_1). Let $\langle \mathcal{M}, \mathcal{N}_0 \rangle$ and $\langle \mathcal{M}, \mathcal{N}_1 \rangle$ be the expansions of $\mathcal{M}$, of respective types $\mathcal{T}_0$ (the type of ψ_0^*) and $\mathcal{T}_1$ (the type of ψ_1^*), given by the induction hypothesis. We can suppose $\mathcal{T}_0 \cap \mathcal{T}_1 = \mathcal{T}$. Let R be the new k-ary relation on $D(\mathcal{M})$, defined as follows: for any k-tuple $\bar{a}_k$ .in $D(\mathcal{M})$,

$$R(\bar{a}_k) \text{ is true iff } \langle \mathcal{M}, \mathcal{N}_1 \rangle \models \forall \bar{y}_d \psi_1^*(\bar{a}_k, \bar{y}_d).$$

Now suppose that $\mathcal{M} \models \varphi(\bar{a}_k)$ for a k-tuple $\bar{a}_k$ in $D(\mathcal{M})$. By the induction hypothesis, we have either $\langle \mathcal{M}, \mathcal{N}_0 \rangle \models \forall \bar{y}_d \psi_0^*(\bar{a}_k, \bar{y}_d)$ or $\langle \mathcal{M}, \mathcal{N}_1 \rangle \models \forall \bar{y}_d \psi_1^*(\bar{a}_k, \bar{y}_d)$, and then as a consequence,

$$\text{(A.1)} \quad \langle \mathcal{M}, \mathcal{N}_0, \mathcal{N}_1, R \rangle \models \forall \bar{y}_d [[R(\bar{a}_k) \vee \psi_0^*(\bar{a}_k, \bar{y}_d)] \wedge [\neg R(\bar{a}_k) \vee \psi_1^*(\bar{a}_k, \bar{y}_d)]].$$

So (i) is proved.

Now we prove (ii). Let $\langle \mathcal{M}, \mathcal{N}_0, \mathcal{N}_1, R \rangle$ be a finite structure of type $\mathcal{T}_0 \cup \mathcal{T}_1 \cup \{R\}$ and let $\bar{a}_k$ be a k-tuple in $D(\mathcal{M})$ for which (A.1) is true. Then according to whether $R(\bar{a}_k)$ is true or false, we have

$$\text{either} \quad \langle \mathcal{M}, \mathcal{N}_1 \rangle \models \forall \bar{y}_d \psi_1^*(\bar{a}_k, \bar{y}_d) \quad \text{or} \quad \langle \mathcal{M}, \mathcal{N}_0 \rangle \models \forall \bar{y}_d \psi_0^*(\bar{a}_k, \bar{y}_d),$$

and so $\mathcal{M} \models \varphi(\bar{a}_k)$.

Case 5. $\varphi(\bar{x}_k) = \exists z \psi(\bar{x}_k, z)$. Let $\mathcal{M}$ be a finite structure of type $\mathcal{T}$ (the type of φ and also of ψ) and let $\langle \mathcal{M}, \mathcal{N} \rangle$ be the expansion of $\mathcal{M}$ of type $\mathcal{T}^*$ (the type of $\psi^*(\bar{x}_k, z, \bar{y}_d)$), given by the induction hypothesis. We construct a k-ary function F on $D(\mathcal{M})$ as follows: for each $\bar{a}_k$ in $D(\mathcal{M})$, $F(\bar{a}_k)$ is a chosen element b such that $\langle \mathcal{M}, \mathcal{N} \rangle \models \forall \bar{y}_d \psi^*(\bar{a}_k, b, \bar{y}_d)$ if there exists one, and if not, then $F(\bar{a}_k)$ is any element in $D(\mathcal{M})$. It is clear that (i) is true with the expansion $\langle \mathcal{M}, \mathcal{N}, F \rangle$. □

Acknowledgments. Many thanks to Pascal Michel for helpful technical discussions. Thanks to Peter Clote for his help.

REFERENCES

[1] A. V. AHO, J. E. HOPCROFT AND J. D. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.
[2] R. V. BOOK AND S. A. GREIBACH, *Quasi-realtime languages*, Math. Systems Theory, 4 (1970), pp. 97–111.
[3] C. C. CHANG AND H. J. KEISLER, *Model Theory*, North-Holland, Amsterdam, 1973.
[4] S. A. COOK, *A hierarchy for nondeterministic time complexity*, J. Comput. Systems Sci., 7 (1973), pp. 343–353.
[5] R. FAGIN, *Generalized first-order spectra and polynomial-time recognizable sets*, in Complexity of Computations, R. M. Karp, ed., American Mathematical Society, Providence, RI, 1974, pp. 43–73.
[6] ———, *A spectrum hierarchy*, Z. Math. Logik Grundlag. Math., 21 (1975), pp. 123–134.
[7] J. HARTMANIS AND R. E. STEARNS, *On the computational complexity of algorithms*, Trans. Amer. Math. Soc., 117 (1965), pp. 285–306.
[8] J. E. HOPCROFT AND J. D. ULLMAN, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading, MA, 1979.
[9] N. IMMERMAN, *Number of quantifiers is better than number of tape cells*, J. Comput. Systems Sci., 22 (1981), pp. 384–405.
[10] ———, *Upper and lower bounds for first-order expressibility*, J. Comput. System Sci., 25, 1 (1982).
[11] N. D. JONES AND A. L. SELMAN, *Turing machines and the spectra of first-order formulas with equality*, J. Symbolic Logic, 39 (1974), pp. 139–150.
[12] H. R. LEWIS, *Complexity results for classes of quantificational formulas*, J. Comput. Systems Sci., 21 (1980), pp. 317–353.
[13] J. F. LYNCH, *Complexity classes and theories of finite models*, Math. Systems Theory, 15 (1982), pp. 127–144.
[14] V. R. PRATT, *Every prime has a succinct certificate*, this Journal, 4 (1975), pp. 214–220.
[15] P. PUDLÁK, *The observational predicate calculus and complexity of computations*, Comment. Math. Univ. Carolin., 16 (1975), pp. 395–398.
[16] ———, Personal communication (1982).
[17] J. I. SEIFERAS, M. J. FISHER AND A. R. MEYER, *Separating nondeterministic time complexity classes*, J. Assoc. Comput. Mach., 25 (1978), pp. 146–167.