Generalized First-Order Spectra and Polynomial-Time Recognizable Sets

Ronald Fagin

Abstract: The *spectrum* of a first-order sentence σ is the set of cardinalities of its finite models. Jones and Selman showed that a set C of numbers (written in binary) is a spectrum if and only if C is in the complexity class NEXP (nondeterministic exponential time). An alternative viewpoint of a spectrum is to consider the spectrum of σ to be the class of finite models of the existential second-order sentence $\exists\mathbf{Q}\sigma(\mathbf{Q})$, where $\mathbf{Q}$ is the similarity type (set of relational symbols) of σ. A *generalized spectrum* is the class of finite models of an existential second-order sentence $\exists\mathbf{Q}\sigma(\mathbf{P},\mathbf{Q})$, where σ is first-order with similarity type $\mathbf{P}\cup\mathbf{Q}$, with $\mathbf{P}$ and $\mathbf{Q}$ disjoint. Let C be a class of finite structures with similarity type $\mathbf{P}$, where C is closed under isomorphism. If $\mathbf{P}$ is nonempty, we show that C is a generalized spectrum if and only if the set of encodings of members of C is in NP. We unify this result with that of Jones and Selman by encoding numbers in unary rather than binary, so that C is a spectrum if and only if C is in NP. We then have that C is a generalized spectrum if and only if the set of encodings of members of C is in NP, whether or not $\mathbf{P}$ is empty. Using this connection between logic and complexity, we take results from complexity theory and convert them into results in logic.

We now mention some of our other results. We show that P = NP if and only if the following apparently much stronger condition holds: there is a constant k such that if T is a "countable" function (a standard notion in automata theory), then every set recognizable nondeterministically in time T can be recognized deterministically in time T^k (analogous to Savitch's Theorem for nondeterministic vs. deterministic space complexity). We show that there is a spectrum S such that $\{n : 2^n \in S\}$ is not a spectrum. In fact, we show that there is such a spectrum S definable using only a single binary relation symbol. This contrasts with the simple result that if S is a spectrum, and if p is a polynomial, then $\{n : p(n) \in S\}$ is a spectrum. Let us say that a generalized spectrum S is *complete* if the following condition holds: the complement of every generalized spectrum is a generalized spectrum if and only if the complement of S is a generalized spectrum. We show that there is a complete generalized spectrum defined by $\exists\mathbf{Q}\sigma(\mathbf{P},\mathbf{Q})$, where $\mathbf{Q}$ consists of a single unary relation symbol, and where $\mathbf{P}$ consists of a single binary relation symbol. W show that if we define a *complete spectrum* similarly, then there is a complete spectrum definable using only a single binary relation symbol. These latter two results are best possible, in terms of minimizing the arity and the number of relation symbols.

From "Complexity of Computation"
(ed. R. Karp)

SIAM–AMS Proceedings
Volume 7
1974

Generalized First-Order Spectra and Polynomial-Time Recognizable Sets[1]

Ronald Fagin

1. Introduction. A *finite structure* is a nonempty finite set, along with certain given functions and relations on the set. For example, a finite group is a set A, along with a binary function $\cdot: A \times A \to A$. If σ is a sentence of first-order logic, then the *spectrum* of σ is the set of cardinalities of finite structures in which σ is true. For example, let σ be the following first-order sentence, where f is a "unary function symbol":

$$\forall x(f(x) \neq x) \wedge \forall x \forall y(f(x) = y \leftrightarrow f(y) = x). \tag{1}$$

Then the spectrum of σ is the set of even positive integers. For, if σ is true about a finite structure $\mathfrak{A} = \langle A; g\rangle$, where A is the universe and g: $A \to A$ (g is the "interpretation" of f), then $\mathfrak{A}$ must look like Figure 1, where $a \to b$ means $g(a) = b$.

$$\begin{array}{c} a_1 \longleftrightarrow a_2 \\ a_3 \longleftrightarrow a_4 \\ a_5 \longleftrightarrow a_6 \\ \vdots \end{array}$$

FIGURE 1

AMS (MOS) subject classifications (1970). Primary 02H05, 68A25; Secondary 02B10, 02E15, 02F10, 68A25, 94A30.

[1] This paper is based on a part of the author's doctoral dissertation in the Department of Mathematics at the University of California, Berkeley. Part of this work was carried out while the author was a National Science Foundation Graduate Fellow and was supported by NSF grant GP-24352.

Copyright © 1974, American Mathematical Society

So, the finite structure $\mathfrak{A}$ has even cardinality. And conversely, for each even positive integer n, there is a way to impose a function on n points to make σ be true about the resulting finite structure.

As a more interesting example, let σ be the conjunction of the field axioms—for example, one conjunct of σ is

$$\forall x \forall y \forall z (x \cdot (y + z) = x \cdot y + x \cdot z).$$

Then the spectrum of σ is the set of powers of primes.

In 1952, H. Scholz [21] posed the problem of characterizing spectra, that is, those sets (of positive integers) which are the spectrum of a sentence of first-order logic. It is well known that every spectrum is recursive: For, assume that we are given a first-order sentence σ and a positive integer n. To determine if n is in the spectrum of σ, we simply systematically write down all finite structures (up to isomorphism) of cardinality n of the relevant type, and test them one by one to see if σ is true in any of them. It is also well known that not every recursive set is a spectrum: We simply form the diagonal set D such that $n \in D$ iff n is not in the nth spectrum (the details are easy to work out).

In 1955, G. Asser [1] posed the problem of whether or not the complement of every spectrum is a spectrum. For example, it is not immediately clear how to write a first-order sentence with spectrum the numbers which are *not* powers of primes.

Note that the spectrum of the sentence (1) is the set of positive integers n for which the following so-called "existential second-order sentence" is true about some (each) set of n points:

$$\exists f (\forall x (f(x) \neq x) \wedge \forall x \forall y (f(x) = y \leftrightarrow f(y) = x)).$$

This suggests a generalization, which is due to Tarski [23]. Let σ be an existential second-order sentence (we will define this and other concepts precisely later), which may have not only bound but free predicate (relation) and function variables. Then the *generalized spectrum* of σ is the class of structures (not numbers) for which σ is true. Let us give some examples. The first few examples will deal with finite structures with a single binary realtion. We can think of these as finite directed graphs.

1. *The class of all k-colorable finite directed graphs, for fixed* $k \geqslant 2$. A (directed) graph $\mathfrak{A} = \langle A; G \rangle$ is *k-colorable* if the universe A of $\mathfrak{A}$ can be partitioned into k subsets $A_1, \cdots, A_k$ such that $\sim Gab$ holds if a and b are in the same subset of the partition. This class is a generalized spectrum, via the following existential second-order sentence, in which Q is a binary predicate symbol which represents the graph relation, and $C_1, \cdots, C_k$ are unary predicate

symbols ($\bigwedge_{i=1}^{k}\phi_i$ abbreviates $\phi_1 \wedge \cdots \wedge \phi_k$; similarly for $\bigvee_{i=1}^{k}\phi_i$):

$$\exists C_1 \cdots \exists C_k\left(\forall x\left(\bigvee_{i=1}^{k} C_ix\right)\wedge\forall x\left(\bigwedge_{i\neq j}\sim(C_ix\wedge C_jx)\right)\right.$$

$$\left.\wedge\forall x\forall y\left(Qxy\longrightarrow\bigwedge_{i=1}^{k}\sim(C_ix\wedge C_iy)\right)\right).$$

2. *The class of finite directed graphs with a nontrivial automorphism.* This class is a generalized spectrum, via the following existential second-order sentence, in which Q is as before, and f is a unary function symbol:

$$\exists f(\exists x(f(x)\neq x)\wedge\forall x\forall y(f(x)=f(y)\longrightarrow x=y)$$

$$\wedge\forall x\forall y(Qxy\leftrightarrow Qf(x)f(y))).$$

3. *The class of finite directed graphs with a Hamilton cycle.* A *cycle* is a finite structure $\langle A;R\rangle$, where A is a set of n distinct elements $a_1,\cdots a_n$ for some n, and $R=\{\langle a_i,a_{i+1}\rangle: 1\leq i<n\}\cup\{\langle a_n,a_1\rangle\}$. A *Hamilton cycle* of $\mathfrak{A}=\langle A;G\rangle$ is a cycle $\langle A;H\rangle$, where $H\subseteq G$. This class is a generalized spectrum, via the existential second-order sentence $\exists<\sigma$, where $<$ is a binary predicate symbol, and where σ is the following first-order sentence (which we translate into English for ease in readability):

> "$<$ is a linear order" $\wedge$ "if y is the immediate successor of x in the linear order, then Qxy" $\wedge$ "if x is the minimum element of the linear order and y the maximum, then Qyx."

Our final example is a class of finite structures with a binary function $\circ$.

4. *The class of nonsimple finite groups.* This class is a generalized spectrum, via

$$\exists N(\text{"the structure is a group"}\wedge\text{"}N\text{ is a nontrivial normal subgroup"}).$$

We can ask the generalized Scholz question, as to how to characterize generalized spectra, and the generalized Asser question, as to whether the complement of every generalized spectrum is a generalized spectrum. Of the examples given, it is easy to see that the non-2-colorable finite directed graphs form a generalized spectrum. It is an open question as to whether the complement of any of the others is a generalized spectrum.

It turns out to be possible to characterize spectra and generalized spectra precisely, in terms of time-bounded nondeterministic Turing machines. The concept of a Turing machine is due, of course, to Turing [24]. The concepts of nondeterministic and multi-tape machines are due to Rabin and Scott [17]. The classification by time complexity is due to Hartmanis and Stearns [12], and by tape complexity, to Hartmanis, Lewis and Stearns [11].

In §§2 and 3, we give definitions and background material. Nothing there is new.

In §4, we show the essential equivalence of generalized spectra and nondeterministic polynomial-time recognizable sets. This supplements the known equivalence of spectra and nondeterministic exponential time recognizable sets of positive integers, which is probably due to James Bennett (unpublished); it was also shown by Jones and Selman [15].

In §5, we show, by analyzing our proof of the automata-theoretic characterization of spectra, that many (all?) spectra are the spectrum of a sentence which has at most one model of each finite cardinality.

In §6, we make use of the automata-theoretic characterization of spectra to show that if spectra are not closed under complement, then a class of candidates for counterexamples suggested by Robert Solovay is sufficient.

In §7, we consider Cook's [7] and Karp's [16] notions of polynomial-completeness and reducibility. We generalize to exponential-completeness, and we directly produce (without making use of Cook's or Karp's results) a polynomial-complete set and an exponential-complete set. This was also done by Book [4]; his sets are similar to ours. We show that completeness implies a certain complement-completeness; using this fact, along with our automata-theoretic characterization of generalized spectra, we show that results in Karp's paper [16] (developed by Karp, Tarjan, and Lawler) give us specific examples of generalized spectra whose complements are generalized spectra iff the complement of every generalized spectrum is a generalized spectrum. In particular, we show that the class of finite directed graphs with a Hamilton cycle is such a "complete" generalized spectrum. Also, we find a complete generalized spectrum defined using only one "extra" (existentialized) unary predicate symbol: This is a best possible result. By making use of automata theory and a result about spectra in the author's doctoral dissertation [9], it is shown that there is a complete spectrum defined using only one binary predicate symbol: This is a best possible result.

In §8, we make use of the polynomial-complete set which we constructed in the previous section to show that if the classes of sets which are polynomial-time recognizable by deterministic and nondeterministic Turing machines are the same, then the following apparently much stronger condition holds: There is a constant k such that essentially any set that can be recognized nondeterministically in time T can be recognized deterministically in time T^k. We then generalize this result in various ways. We conclude §8 by an analogy with Post's problem.

In §9, we make use of a tape-complexity argument similar to one used by Bennett [2] to show that there is a spectrum S such that $\{n: 2^n \in S\}$ is not a spectrum. By making use of a result in [9], we then show that there is such a spectrum S defined using only one binary predicate symbol. We also show that our techniques give a new proof of a theorem of Book [3], that the two classes of sets recognizable nondeterministically in polynomial time or in exponential time respectively are different.

In §10, we exhibit an example of a polynomial-complete set which is recognized by a nondeterministic two-tape Turing machine in real time. The existence of such a set follows immediately from theorems of Hunt [14], and of Book and Greibach [5].

2. Notions from logic. Denote the set of *positive integers* $\{1, 2, 3, \cdots\}$ by Z^+, and the set $\{0, \cdots, n-1\}$ by n. By the *natural numbers* we mean the set $Z^+ \cup \{0\}$. If A is a set, then card A is the cardinality of the set. Denote the set of k-tuples $\langle a_1, \cdots, a_k\rangle$ of members of A by A^k.

A *finite similarity type* is a finite set of predicate symbols and function symbols. Each predicate symbol (function symbol) has a positive integer (natural number), the *degree*, associated with it. If a symbol has degree k, then call the symbol k-ary. We will often call 1-ary symbols *unary*, and 2-ary symbols *binary*. A constant symbol is a 0-ary function symbol. We will denote finite similarity types by the letters S and T.

Assume that S contains the n distinct symbols $Q_1, \cdots, Q_n$, written in some fixed order. Then a *finite S-structure* $\mathfrak{A}$ is an $(n+1)$-tuple $\langle A; R_1, \cdots, R_n\rangle$ (where we write a semicolon after the first member), such that we have the following:

1. A is a nonempty finite set, called the *universe* (of $\mathfrak{A}$), and denoted $|\mathfrak{A}|$.
2. If Q_i is a k-ary predicate symbol, then R_i is a subset of A^k.
3. If Q_i is a k-ary function symbol, and $k > 0$, then R_i is a function from A^k into A.
4. If Q_i is a constant symbol, then $R_i \in A$.

In each case, write $R_i = Q_i^{\mathfrak{A}}$. We will sometimes make use of a *graph predicate symbol* Q; if $Q \in S$, then, for $\mathfrak{A}$ to be a finite S-structure, $Q^{\mathfrak{A}}$ must be a graph (i.e., irreflexive and symmetric), or, equivalently, a set of unordered pairs (of members of $|\mathfrak{A}|$). Denote the cardinality of $|\mathfrak{A}|$ by card $(\mathfrak{A})$. Denote the class of finite S-structures by Fin(S); abbreviate Fin$(\{Q_1, \cdots, Q_n\})$ by Fin$(Q_1, \cdots, Q_n)$.

Assume that S and T are disjoint finite similarity types, that $\mathfrak{A}$ is a finite $S \cup T$-structure, and that $\mathfrak{B}$ is a finite S-structure. Then $\mathfrak{A}$ is an *expansion* of $\mathfrak{B}$ (to $S \cup T$) if $|\mathfrak{A}| = |\mathfrak{B}|$ and $Q^{\mathfrak{A}} = Q^{\mathfrak{B}}$ for each Q in S. We write $\mathfrak{B} = \mathfrak{A} \upharpoonright S$.

The metamathematical language we will be working in is a set of symbols $\sim$, $\wedge$, $\forall$, $=$; an infinite number of individual variables u, v, w, x, y, z along with affixes; the left and right parentheses (,); and predicate and function variables. We do not distinguish between predicate or function *symbols* and predicate or function *variables*. Except in this section, whenever we refer to a variable, we will always mean an individual variable.

A *term* is a member of the smallest set T which contains the 0-ary function variables and the individual variables, and which contains $f(t_1, \cdots, t_k)$ for each k-ary function variable f and each $t_1, \cdots, t_k$ in T.

An *atomic formula* is an expression $t_1 = t_2$ or $Qt_1 \cdots t_k$, where the t_i are terms and Q is a k-ary predicate variable. A *first-order formula* is a member of the smallest set which contains each atomic formula, and which contains $\sim \phi_1, (\phi_1 \wedge \phi_2)$, and $\forall x \phi_1$ (for each individual variable x), whenever it contains ϕ_1 and ϕ_2. A *second-order formula* is a member of the smallest set which contains each atomic formula, and which contains $\sim \phi_1, (\phi_1 \wedge \phi_2), \forall x \phi_1$ (for each individual variable x) and $\forall Q \phi_1$ (for each predicate or function variable Q) whenever it contains ϕ_1 and ϕ_2.

The formulas $\phi_1 \vee \phi_2$, $\exists x \phi$, $(\exists x \neq y)\phi$, $\exists! x \phi$ (read "there exists exactly one x such that ϕ"), and so on, are defined in the usual way, e.g., $\phi_1 \vee \phi_2$ is $\sim(\sim \phi_1 \wedge \sim \phi_2)$. If $T = \{Q_1, \cdots, Q_n\}$ is a finite similarity type, then $\exists T \phi$ is $\exists Q_1 \cdots \exists Q_n \phi$. If ϕ is a first-order formula, then $\exists T \phi$ is called an *existential second-order formula*.

If $x_1, \cdots, x_m$ are individual variables, then we will sometimes write $\mathbf{x}$ as an abbreviation for the m-tuple $\langle x_1, \cdots, x_m \rangle$, when this will lead to no confusion. We may write $\forall \mathbf{x} \phi$ for $\forall x_1 \cdots \forall x_m \phi$.

The notion of a variable being a *free variable* is understood in the usual way. Let S be a fixed finite similarity type. An *S-formula* is a (first- or second-order) formula all of whose free predicate and function variables are in S. A *sentence* is a formula with no free individual variables. A formula is *quantifier-free* if it contains no quantifiers ($\forall$ or $\exists$).

Assume that $\mathfrak{A}$ is a finite S-structure, and that σ is a first- or second-order S-sentence. Then $\mathfrak{A} \models \sigma$ means that σ is true in $\mathfrak{A}$; we say that $\mathfrak{A}$ *is a model of* σ. For a precise definition of truth, see [22]. We note that the equality symbol $=$ is always given the standard interpretation. We define $\mathrm{Mod}_\omega \sigma$ to be the class of all finite S-structures which are models of σ.

Assume that S and T are disjoint finite similarity types, and that $A \subseteq \mathrm{Fin}(S)$. Then A is an *S-spectrum*, or an *(S, T)-spectrum*, if there is a first-order $S \cup T$-sentence σ such that $A = \mathrm{Mod}_\omega \exists T\sigma$. This is simply Tarski's [23] notion of PC, in the special case where we restrict to the class of finite structures. A *generalized spectrum* is an S-spectrum for some S. A *monadic generalized spectrum* is an (S, T)-spectrum where T is a set of unary predicate symbols. A *spectrum* is an S-spectrum for S empty; if A is a spectrum, then we identify $\{n: \langle n\rangle \in A\} \subseteq Z^+$ with A. In this case, if $A = \{n: \langle n\rangle \models \exists T\sigma\}$, then we call A the *spectrum of* σ.

3. **Notions from automata theory.** When A is a finite set of symbols, then A^* is the set of *strings* or *words*, that is, the finite concatenations $a_1 \frown a_2 \frown \cdots \frown a_n$ of members of A. The *length* of $a = a_1 \frown \cdots \frown a_n$ is n (written $\mathrm{len}(a) = n$). If $k \in Z^+$, then $\mathrm{len}(k)$ is the length of the binary representation of k; this corresponds to a convention that we will always represent positive integers in binary notation. If a set $S \subseteq A^*$ for some finite set A, then S is a *language*.

An *m-tape nondeterministic Turing machine* M is an 8-tuple $\langle K, \Gamma, B, \Sigma, \delta, q_0, q_A, q_R\rangle$, where K is a finite set (the *states* of M); Γ is a finite set (the *tape symbols* of M); B is a member of S (the *blank*); Σ is a subset of $(\Gamma - \{B\})$ (the *input symbols* of M); q_0, q_A, and q_R are members of K (the *initial state*, *accepting final state*, and *rejecting final state* of M, respectively); and δ is a mapping from $(K - \{q_A, q_R\}) \times \Gamma^m$ to the set of nonempty subsets of $K \times (\Gamma - \{B\})^m \times \{L, R\}^m$ (the table of *transitions, moves*, or *steps* of M).

If the range of δ consists of singletons sets, that is, sets with exactly one member, then M is an *m-tape deterministic Turing machine.*

We may sometimes call M simply a *machine.*

An *instantaneous description* of M is a $(2m + 1)$-tuple $I = \langle q; \alpha^1, \cdots, \alpha^m; i_1, \cdots, i_m\rangle$, where $q \in K$, where $\alpha^j \in (\Gamma - \{B\})^*$, and where $1 \leqslant i_j \leqslant \mathrm{len}(\alpha^j) + 1$, for $1 \leqslant j \leqslant m$. We say that M is in state q, that α^j is the nonblank portion of the jth tape, and that the jth tape head is scanning $(\alpha^j)_{i_j}$, the i_jth symbol of the word α^j (or that M is scanning $(\alpha^j)_{i_j}$ on the jth tape); we also say that the jth tape head is scanning the i_jth tape square.

Let $I' = \langle q'; \alpha^{1\prime}, \cdots, \alpha^{m\prime}; i_1', \cdots, i_m'\rangle$ be another instantaneous description of M. We say that $I \to_M I'$ if $q \neq q_A$, $q \neq q_R$, and if there is $s = \langle p; a_1, \cdots, a_m; T_1, \cdots, T_m\rangle$ in $\delta(q; (\alpha^1)_{i_1}, \cdots, (\alpha^m)_{i_m})$ such that $p = q'$, and, for each j, with $1 \leqslant j \leqslant m$:

1. $(\alpha^{j\prime})_{i_j} = a_j$.
2. $(\alpha^{j\prime})_k = (\alpha^j)_k$ for $1 \leqslant k \leqslant \mathrm{len}(\alpha^j)$, if $k \neq i_j$.
3. $\mathrm{len}(\alpha^{j\prime}) = \mathrm{len}(\alpha^j)$ unless $i_j = \mathrm{len}(\alpha^j) + 1$; in that case, $\mathrm{len}(\alpha^{j\prime}) = \mathrm{len}(\alpha^j) + 1$.

4. If $T_j = L$, then $i_j \neq 1$.

5. If $T_j = R$, then $i'_j = i_j + 1$; if $T_j = L$, then $i'_j = i_j - 1$.

We say that M prints a_j on the jth tape. Note that M cannot print a blank (that is, $a_j \neq B$); so, we say that α^j is that portion of the jth tape which has been visited, or scanned. If $T_j = R\,(L)$, then we say that the jth tape head moves to the right (left). Assumption 4 corresponds to the intuitive notion of each tape being one-way infinite to the right; thus, if M "orders a tape head to go off the left end of its tape," then M halts. It is important to observe that it is possible to have $I \longrightarrow_M I_1$ and $I \longrightarrow_M I_2$ with $I_1 \neq I_2$; hence the name "nondeterministic."

We say $I \longrightarrow_M^* J$ if there is a finite sequence $I_1, \cdots, I_n$ such that $I_1 = I$, $I_n = J$, and $I_i \longrightarrow_M I_{i+1}$ for $1 \leqslant i < n$. Denote the empty word in Σ^* by Λ. If $w \in \Sigma^*$, then let $\bar{w} = \langle q_0; w, \Lambda, \cdots, \Lambda; 1, \cdots, 1\rangle$ (w is the *input*). Call an instantaneous description $\langle q; \alpha^1, \cdots, \alpha^m; i_1, \cdots, i_m\rangle$ *accepting* (*rejecting*) if $q = q_A$ ($q = q_R$). We say that M *accepts* w in Σ^* if $\bar{w} \longrightarrow_M^* I$ for some accepting I. Denote by A_M the set of all words accepted by M. We say that M *recognizes* A_M.

If $\bar{w} \longrightarrow_M^* I$ for some accepting (rejecting) I, then we say that M, with w as input, eventually enters the accepting (rejecting) final state, and halts.

Intuitively speaking, there are three ways that a word w in Σ^* may be not accepted by M: M, with w as input, can eventually enter the rejecting final state q_R; or M can order a tape head to go off the left end of its tape; or M can never halt.

Assume that M is a multi-tape nondeterministic Turing machine, $w \in A_M$, and t is a positive integer. We say that M *accepts* w *within* t *steps* if, for some $n \leqslant t$,

(2) there are instantaneous descriptions $I_1, \cdots, I_{n+1}$ such that $I_1 = \bar{w}$, I_{n+1} is accepting, and $I_k \longrightarrow_M I_{k+1}$ for $1 \leqslant k \leqslant n$.

Let s be a positive integer. Then M *accepts* w *within space* s if for some positive integer n, (2) holds and, for each I_k, $1 \leqslant k \leqslant n+1$, if $I_k = \langle q; \alpha^1, \cdots, \alpha^m; i_1, \cdots, i_m\rangle$, then $i_p \leqslant s$ for $1 \leqslant p \leqslant m$.

Let $T\colon N \longrightarrow N$ and $S\colon N \longrightarrow N$ be functions. We say that M *operates in time* T (*tape* S), or M *recognizes* A_M *in time* T (*tape* S) if, for each natural number l and each word w in A_m of length l, the machine M accepts w within $T(l)$ steps (space $S(l)$). We say that A *is recognizable* (*non*)*deterministically in time* T, or *tape* S, if there is a multi-tape (non)deterministic Turing machine M that operates in time T, or tape S, such that $A = A_M$.

We will now define some well-known, important classes. Let P (NP) be the class of sets A for which there is a positive integer k such that A is recognizable (non)deterministically in time $l \mapsto l^k$. These are the *(non)deterministic polynomial-time recognizable sets.*

Let P_1 (NP_1) be the class of sets A for which there is a positive integer k such that A is recognizable (non)deterministically in time $l \mapsto 2^{kl}$. These are the *(non)deterministic exponential-time recognizable sets.* If the positive integer n has length l in binary notation, then $2^{l-1} \leq n < 2^l$. Therefore, a set A of positive integers is in P_1 (NP_1) iff there is a multi-tape (non)deterministic Turing machine M, and a positive integer k such that $A = A_M$ and M accepts each n in A within n^k steps. So in some sense, P_1 and NP_1 are also classes of polynomial time recognizable sets.

We say that a set A is recognizable in *real time* if A is recognizable in time $l \mapsto l + 1$. We use $l + 1$ instead of l, so that the machine can tell when it reaches the end of the input word.

We have defined Turing machines which recognize sets rather than compute functions. It is clear how to modify our definitions to get the usual notion of a function f computable by a deterministic one-tape Turing machine M; it is also clear what we mean by *M computes the value of f at w within t steps.* If $f: A \to B$, where A and B are languages, and if $T: N \to N$, then we say that *M computes f in time T* if, for each natural number l and each word w in A of length l, the machine M computes the value of f at w within $T(l)$ steps. We define Π to be the class of functions which are computable by a one-tape deterministic Turing machine in polynomial time. Functions are generally considered easy to compute if they are in Π; Cobham [6] was the first to single out this class. We define Π_1 to be the class of functions f which are computable by a one-tape deterministic Turing machine in exponential time, and for which there is a constant c such that $\text{len}(f(w)) \leq c \cdot \text{len}(w)$ for each w in the domain of f.

We now state without proof two theorems, which were essentially proved in [12]. The proofs can also be found in [13, pp. 139–140 and 143].

THEOREM 1. *If A is recognized by a one-tape (non)deterministic Turing machine in time T, if $\liminf_{l\to\infty} T(l)/l^2 = \infty$, and if $c > 0$ is arbitrary, then A is recognized by a one-tape (non)deterministic Turing machine in time $l \mapsto \max(l + 1, cT)$.*

THEOREM 2. *If A is recognized by an m-tape (non)deterministic Turing machine in time T, and if $\liminf_{l\to\infty} T(l)/l = \infty$, then A is recognized by a one-tape (non)deterministic Turing machine in time T^2.*

It follows from Theorem 2 that the concepts of polynomial and exponential time are invariant, whether we consider one-tape or multi-tape machines.

A function T is *countable* if there is a positive integer c and a two-tape deterministic Turing machine that operates in time cT which, for each natural number l and each word of length l as input (on the first tape), halts (by entering a final state) with a string of at least $T(l)$ tallies on its second tape (a tally is a one). This is slightly broader than the usual definition, but more convenient for us to use. We will make use of the fact that $l \longmapsto l^k$ is countable for each positive integer k; for, l^k can be calculated in a polynomial of $\mathrm{len}(l)$ time, which is less than l for sufficiently large l.

A *linear-bounded automaton* is a one-tape deterministic Turing machine that operates in tape $l \longmapsto l + 1$. We denote by $\mathcal{E}^2_*$ those subsets of Z^+ whose characteristic functions are in the Grzegorczyk class $\mathcal{E}^2$ [10]. The class $\mathcal{E}^2$ is the smallest class which contains the successor and multiplication functions, and is closed under explicit transformation, composition, and limited recursion. We are interested in the class $\mathcal{E}^2_*$ precisely because of the following theorem.

THEOREM 3 (RITCHIE [18]). *A set of positive integers is recognizable by a linear-bounded automaton iff it is in* $\mathcal{E}^2_*$.

We will make use of the following well-known simple theorem, which we state without proof.

THEOREM 4. *The classes* $\mathcal{E}^2_*$, $\mathcal{P}$, *and* $\mathcal{P}_1$ *are closed under complement.*

A function $S: N \longrightarrow N$ is said to be *constructible* if there is a one-tape deterministic Turing machine which operates in tape S, but not in tape S', if $S'(l) < S(l)$ for some l. We conclude this section by stating a theorem which is essentially proved in [11]. The proof can also be found in [13, pp. 150–151].

THEOREM 5. *Assume that* S *is a constructible tape function with* $S(l) \geqslant \log_2 l$ *for each natural number* l. *Then there is a set which is recognizable by a one-tape deterministic Turing machine in tape* S, *but which is not recognizable in tape* S' *for any function* S' *with* $\liminf_{l \to \infty} S'(l)/S(l) = 0$.

4. Generalized spectra and automata. In this section, we will prove the theorem (Theorem 6) which interrelates spectra and generalized spectra with automata.

Let S be a fixed finite similarity type which (for convenience) contains only predicate symbols, and let $P_1, \cdots, P_r$ be the predicate symbols in S in some fixed order. Let $\Sigma = \{0, 1, \#\}$.

Assume that $\mathfrak{A} = \langle\{1, \cdots, n\}; S_1, \cdots, S_r\rangle$ is a finite S-structure, and that P_i (and hence S_i) is m_i-ary ($1 \leq i \leq r$). For each i, define b_i to be the word in $\{0, 1\}^*$ of length n^{m_i} such that if $\langle c_1, \cdots, c_{m_i} \rangle$ is the kth member of $\{1, \cdots, n\}^{m_i}$ in lexicographical order, then the kth digit of b_i is 1 if $S_i c_1 \cdots c_{m_i}$ and 0 otherwise ($1 \leq k \leq n^{m_i}$). Let $e(\mathfrak{A})$, the *encoding* of $\mathfrak{A}$, be the word $a \# b_1 \# b_2 \# \cdots \# b_r$ in Σ^*, where a is the binary representation of n. If A is a class of finite S-structures, then let $E(\mathsf{A}) = \{e(\mathfrak{A})\colon (\exists n \in Z^+) (|\mathfrak{A}| = \{1, \cdots, n\}$ and $\mathfrak{A} \in \mathsf{A})\}$.

THEOREM 6. *Assume that* $\mathsf{A} \subseteq \mathrm{Fin}(S)$, *and that* A *is closed under isomorphism.*

1. *If* $S \neq \emptyset$, *then* A *is an* S-*spectrum iff* $E(\mathsf{A}) \in NP$.
2. *If* $S = \emptyset$, *then* A *is a spectrum iff* $E(\mathsf{A}) \in NP_1$.

Note. We can combine these last two statements by saying that A is an S-spectrum ($S = \emptyset$ or $S \neq \emptyset$) iff there is a positive integer k and there is a nondeterministic multi-tape Turing machine which recognizes $E(\mathsf{A})$, and which accepts each $e(\mathfrak{A})$ in $E(\mathsf{A})$ within n^k steps, where $|\mathfrak{A}| = \{1, \cdots, n\}$.

PROOF. Assume that A is an S-spectrum (possibly $S = \emptyset$.) Then, for some positive integers t and k, some set T of t new k-ary predicate symbols, and some first-order $S \cup T$-sentence $\sigma = Q_1 x_1 \cdots Q_m x_m \phi$, where ϕ is quantifier-free and each Q_i is $\forall$ or $\exists$, we have $\mathsf{A} = \mathrm{Mod}_\omega \exists T\sigma$. This is because of well-known techniques of simulating $(k-1)$-ary functions and $(k-1)$-ary relations by k-ary relations, and because each first-order sentence is equivalent to a sentence with all quantifiers out front (so-called "prenex normal form").

We will informally describe a $(t + m + 2)$-tape nondeterministic Turing machine M which recognizes $E(\mathsf{A})$. The first tape is the input tape. The machine M first tests to see if the input is of form $a \# b_1 \# b_2 \# \cdots \# b_r$, with a in $\{0, 1\}^*$ and starting with a 1, with r the number of (predicate) symbols in S, and with each b_i in $\{0, 1\}^*$ and of the proper length; to test for proper length, M uses its last tape as a "counter." If the input is not of the proper form, then M rejects. If the input is of the proper form, then say the input is $e(\mathfrak{A})$, and $|\mathfrak{A}| = \{1, \cdots, n\}$. On each of the 2nd through $(t + 1)$st tapes, M then nondeterministically prints a string of n^k 0's and 1's, by using the last tape as a counter; these correspond to "guesses" for the interpretations of the predicate symbols in T. Let $\mathfrak{A}'$ be the obvious expansion of $\mathfrak{A}$ to $S \cup T$.

Next, on the $(t + i + 1)$st tape, M systematically prints each possibility a_i for x_i ($1 \leq i \leq m$), where a_i runs between 1 and n. There are n^m possibilities for the m-tuple $\langle a_1, \cdots, a_m \rangle$. For each given such possibility, M can easily test to see if $\phi(a_1, \cdots, a_m)$ holds in $\mathfrak{A}'$, where it again makes use of the last tape as a work tape. It is easy to see how to arrange the logic to test whether $\mathfrak{A}' \models \sigma$.

So M recognizes $E(A)$, and there is a nonnegative polynomial p such that M accepts each $e(\mathfrak{A})$ in $E(A)$ nondeterministically within $p(n)$ steps, where $|\mathfrak{A}| = \{1, \cdots, n\}$. Let l be the length of the input $e(\mathfrak{A})$. If $S = \emptyset$, then n is approximately 2^l. If $S \neq \emptyset$, then l is approximately tn^k (in each case, "approximately" means up to a fixed constant factor). So if $S = \emptyset$ then $E(A)$ can be recognized nondeterministically in time $l \longmapsto p(2^l)$, and hence $E(A) \in NP_1$. If $S \neq \emptyset$, then $E(A)$ can certainly be recognized nondeterministically in time $l \longmapsto p(l)$, and hence $E(A) \in NP$.

Conversely, assume that $E(A)$ is in NP or NP_1, depending on whether $S \neq \emptyset$ or $S = \emptyset$. Assume that $S = \{P_1, \cdots, P_r\}$, where P_i is m_i-ary, for $1 \leq i \leq r$. It will be convenient to define a slightly modified $(r + 1)$-tape nondeterministic Turing machine M, by changing the definition of an input. If x is an $(r + 1)$-tuple $\langle a_1, \cdots, a_{r+1}\rangle$, then let $\bar{x} = \langle q_0; a_1, \cdots, a_{r+1}; 1, \cdots, 1\rangle$; we say that M accepts the $(r + 1)$-tuple x if $\bar{x} \longrightarrow_M^* I$ for some accepting instantaneous description I.

It is clear that there is a positive integer k and a modified $(r + 1)$-tape nondeterministic Turing machine M which accepts precisely those $(r + 1)$-tuples $\langle a', b_1, \cdots, b_r\rangle$ such that $a \# b_1 \# b_2 \# \cdots \# b_r$ is in $E(A)$, where a' is the string a written backwards, and such that M accepts such an input within n^k steps, where n is the number represented by a in binary notation. We can assume that $k \geq \max \{m_i: 1 \leq i \leq r\}$. It is clear that if M accepts $\langle a', b_1, \cdots, b_r\rangle$ within n^k steps, then it accepts $\langle a', b_1, \cdots, b_r\rangle$ within space n^k; we will make use of this fact.

Introduce the set T of the following new symbols. The symbol $<$ is a binary predicate symbol, which represents a linear order; c_1 and c_2 are constant symbols, which represent respectively the minimal and maximal members of the linear order; 0, 1, and B are constant symbols, which represent respectively the zero, one and blank tape symbols; q_0, q_A, and q_R are constant symbols, which represent respectively the initial state, accepting final state, and rejecting final state; S is a unary function symbol, which represents successor in the linear order $<$; S_1 is a $2k$-ary predicate symbol, where

$$S_1(x_1, \cdots, x_k; y_1, \cdots, y_k)$$

means that $\mathbf{y}$ is the successor of $\mathbf{x}$ in the lexicographical ordering; q is a k-ary function symbol, where $q(t_1, \cdots, t_k)$ is the state that the machine is in at time $\mathbf{t}$; v_i is a $2k$-ary function symbol, for $1 \leq i \leq r + 1$, where $v_i(t_1, \cdots, t_k; x_1, \cdots, x_k)$ is the tape symbol printed on square $\mathbf{x}$ of the ith tape at time $\mathbf{t}$; H_i is a $2k$-ary predicate symbol, for $1 \leq i \leq r + 1$, where $H_i(\mathbf{t}; \mathbf{x})$ means that, at time $\mathbf{t}$, the ith tape head is scanning square $\mathbf{x}$ on the

ith tape; and G is a binary function symbol, where $G(x, i)$ is the ith digit from the right in the binary representation of x, if we think of the binary representation of the positive integers less than or equal to n (the cardinality of the universe) as being given by a word of length n, which starts out with a series of blanks, followed by the usual binary representation.

We think of the k-tuple $\langle c_1, \cdots, c_1\rangle$ as representing the first time unit (and the first tape square on each tape); if $S_1(\mathrm{x}; \mathrm{y})$, then y is the next time unit (next tape square) after x. Thus, the k-tuple $\langle c_2, \cdots, c_2\rangle$ represents the n^kth time unit (n^kth tape square).

Assume that Γ contains g tape symbols. We represent these by c_1, $S(c_1)$, $S(S(c_1)), \cdots, S^{(g-1)}(c_1)$, where c_1 represents the zero, $S(c_1)$ represents the one, and $S^{(2)}(c_1)$ represents the blank. For ease in readability, we have introduced the symbols 0, 1, and B, which will denote the same elements (in a model) as c_1, $S(c_1)$, and $S^{(2)}(c_1)$ respectively. Assume that K contains p states. We represent these by $c_1, \cdots, S^{(p-1)}(c_1)$ where, for ease in readability, we have q_0, q_A, and q_R denoting the same elements as c_1, $S(c_1)$, and $S^{(2)}(c_1)$ respectively.

Let σ_1 be the conjunction of the following sentences:

$$\begin{array}{ll} 0 = c_1, & q_0 = c_1, \\ 1 = S(c_1), & q_A = S(c_1), \\ B = S^{(2)}(c_1), & q_R = S^{(2)}(c_1). \end{array}$$

Let σ_2 be the sentence "$<$ is a linear order, c_1 is minimal, c_2 is maximal, and S is successor, except $S(c_2) = c_1$."

Let σ_3 be the sentence which says that $S_1(\mathrm{x}; \mathrm{y})$ holds iff y is the successor of x in lexicographical order, except that $S_1(c_2, \cdots, c_2; c_1, \cdots, c_1)$ holds. Thus, σ_3 is the conjunction of the following $k+2$ sentences:

$$\forall x_1 \cdots \forall x_k \exists ! y_1 \cdots \exists ! y_k S_1(x_1, \cdots, x_k; y_1, \cdots, y_k),$$

$$\forall x_1 \cdots \forall x_k (x_k \neq c_2 \longrightarrow S_1(x_1, \cdots, x_k; x_1, \cdots, x_{k-1}, Sx_k)),$$

$$\forall x_1 \cdots \forall x_k ((x_k = c_2 \wedge x_{k-1} \neq c_2)$$
$$\longrightarrow S_1(x_1, \cdots, x_k; x_1, \cdots, x_{k-2}, Sx_{k-1}, c_1)),$$
$$\cdots$$
$$S_1(c_2, \cdots, c_2; c_1, \cdots, c_1).$$

The conjunction σ_4 of the following sentences defines G to be what we said we wanted:

$$G(c_1, c_1) = 1,$$

$$(\forall x \neq c_1)(G(c_1, x) = B),$$

$$(\forall x \neq c_2)\forall y(((\exists z < y)(G(x, z) = 0 \vee G(x, z) = B)) \longrightarrow (G(Sx, y) = G(x, y))),$$

$$(\forall x \neq c_2)\forall y(((\exists z < y)(G(x, z) = 1 \wedge G(x, y) = 0)) \longrightarrow (G(Sx, y) = 1)),$$

$$(\forall x \neq c_2)\forall y(((\forall z < y)(G(x, z) = 1 \wedge G(x, y) = 1)) \longrightarrow (G(Sx, y) = 0)),$$

$$(\forall x \neq c_2)\forall y(((\forall z < y)(G(x, z) = 1 \wedge G(x, y) = B)) \longrightarrow (G(Sx, y) = 1)).$$

The conjunction σ_5 of the following sentences gives self-explanatory information about q and the H_i:

$$q(c_1, \cdots, c_1) = q_0,$$

$$q(c_2, \cdots, c_2) = q_A,$$

$$\bigwedge_{i=1}^{r+1} \forall t_1 \cdots \forall t_k \exists! x_1 \cdots \exists! x_k H_i(\mathrm{t}; \mathrm{x}),$$

$$\bigwedge_{i=1}^{r+1} H_i(c_1, \cdots, c_1; c_1, \cdots, c_1).$$

The conjunction σ_6 of the next two sentences initializes the first tape so that it starts with the binary representation of n (the cardinality of the universe) running backwards, followed by blanks:

$$\forall x(v_1(c_1, \cdots, c_1; c_1, \cdots, c_1, x) = G(c_2, x)),$$

$$\forall x_1 \cdots \forall x_k(\sim (x_1 = c_1 \wedge \cdots \wedge x_{k-1} = c_1) \longrightarrow (v_1(c_1, \cdots, c_1; x_1, \cdots, x_k) = B)).$$

The conjunction σ_7 of the following sentences initializes the 2nd through $(r+1)$st tapes such that the $(i+1)$st tape starts out with a string of 0's and 1's which represents P_i, followed by blanks $(1 \leq i \leq r)$:

$$\bigwedge_{i=1}^{r} \forall x_1 \cdots \forall x_{m_i}(P_i x_1 \cdots x_{m_i} \rightarrow (v_{i+1}(c_1, \cdots, c_1; c_1, \cdots, c_1, x_1, \cdots, x_{m_i}) = 1)),$$

$$\bigwedge_{i=1}^{r} \forall x_1 \cdots \forall x_{m_i}(\sim P_i x_1 \cdots x_{m_i} \rightarrow (v_{i+1}(c_1, \cdots, c_1; c_1, \cdots, c_1, x_1, \cdots, x_{m_i}) = 0)),$$

$$\bigwedge_{i=1}^{r} \forall x_1 \cdots \forall x_k(\sim (x_1 = c_1 \wedge \cdots \wedge x_{k-m_i} = c_1) \rightarrow (v_{i+1}(c_1, \cdots, c_1; x_1, \cdots, x_k) = B)).$$

The sentence σ_8 says that after the machine enters a final state, nothing ever changes. Here $\mathbf{u}$ represents the next time unit after $\mathbf{t}$:

$$\forall \mathbf{t} \forall \mathbf{u}\Bigg(\sim (t_1 = c_2 \wedge \cdots \wedge t_k = c_2) \wedge S_1(\mathbf{t}; \mathbf{u}) \wedge (q(\mathbf{t}) = q_A \vee q(\mathbf{t}) = q_R) \rightarrow \Bigg((q(\mathbf{u}) = q(\mathbf{t})) \wedge \forall \mathbf{x}\left(\bigwedge_{i=1}^{r+1} (v_i(\mathbf{u}; \mathbf{x}) = v_i(\mathbf{t}; \mathbf{x}))\right) \wedge \forall \mathbf{x} \bigwedge_{i=1}^{r+1} (H_i(\mathbf{u}; \mathbf{x}) \leftrightarrow H_i(\mathbf{t}; \mathbf{x}))\Bigg)\Bigg).$$

The sentence σ_9 is a conjunction of sentences which describe the table of transitions of M, entry by entry. Assume that $\delta(b; e_1, \cdots, e_{r+1}) = \{s_1, \cdots, s_w\}$, that we are representing the state b by $S^{(d)}(c_1)$, and that we are representing the tape symbol e_i by $S^{(f_i)}(c_1)$, for $1 \leqslant i \leqslant r$. Then one conjunct of σ_9 is the following sentence:

$$\forall \mathbf{t} \forall \mathbf{u} \forall x_1^1 \cdots \forall x_k^1 \cdots \forall x_1^{r+1} \cdots \forall x_k^{r+1} \Bigg(\sim (t_1 = c_2 \wedge \cdots \wedge t_k = c_2) \wedge S_1(\mathbf{t}; \mathbf{u}) \wedge \bigwedge_{i=1}^{r+1} H_i(\mathbf{t}; x_1^i, \cdots, x_k^i) \wedge (q(\mathbf{t}) = S^{(d)}(c_1)) \wedge \bigwedge_{i=1}^{r+1} (v_i(\mathbf{t}; x_1^i, \cdots, x_k^i) = S^{(f_i)}(c_1)) \rightarrow \bigvee_{i=1}^{w} \phi_i\Bigg),$$

where ϕ_i tells the transition which is possible, according to s_i, for $1 \leqslant i \leqslant w$.

Specifically, assume that s_i is $\langle a; b_1, \cdots, b_{r+1}; T_1, \cdots, T_{r+1}\rangle$, where we are representing the state a by $S^{(m)}(c_1)$, where we are representing the symbol b_j by $S^{(d_j)}(c_1)$, for $1 \leq j \leq r+1$, and where each T_j is either R or L. Let $I = \{j: T_j = R\}$, and $J = \{j: T_j = L\}$. Then ϕ_i is the conjunction of the following formulas, where in the last conjunction we include the restriction that no tape head go off the left end of its tape:

$$q(\mathrm{u}) = S^{(m)}(c_1),$$

$$\bigwedge_{j=1}^{r+1} \forall \mathrm{z}(\sim (z_1 = x_1^j \wedge \cdots \wedge z_k = x_k^j) \longrightarrow (v_j(\mathrm{u}; \mathrm{z}) = v_j(\mathrm{t}; \mathrm{z}))),$$

$$\bigwedge_{j=1}^{r+1} v_j(\mathrm{u}; \mathrm{x}^j) = S^{(d_j)}(c_1), \qquad \bigwedge_{j\in I} \forall \mathrm{y}^j(S_1(\mathrm{x}^j; \mathrm{y}^j) \longrightarrow H_j(\mathrm{u}; \mathrm{y}^j)),$$

$$\bigwedge_{j\in J} (\sim (x_1^j = c_1 \wedge \cdots \wedge x_k^j = c_1) \wedge \forall \mathrm{y}^j(S_1(\mathrm{y}^j; \mathrm{x}^j)) \longrightarrow H_j(\mathrm{u}; \mathrm{y}^j)).$$

If $n \geq \max(\text{card}\,\Gamma, \text{card}\,K)$, then an S-structure $\mathfrak{A}$ with card $(\mathfrak{A}) = n$ is in A iff $\mathfrak{A} \models \exists \mathrm{T}(\bigwedge_{i=1}^{9} \sigma_i)$. It is well known that each "finite modification" of an S-spectrum is an S-spectrum. Therefore, A is an S-spectrum. □

Apparently, James Bennett was the first to prove part 2 of Theorem 6, although he did not publish it. The first published proof (a different proof from ours) is by Jones and Selman [15]. Part 1 is new.

It is fairly easy to prove from Theorem 6 that

(3) the class of (generalized) spectra is closed under complement iff NP_1 (NP) is closed under complement.

This is because there are not only simple ways to encode finite structures into strings of symbols, but also ways to "encode" strings of symbols into finite structures. We will not demonstrate this, because (3) follows easily from our work on complete sets in §7.

We know from Theorem 4 that P_1 (P) is closed under complement. So if $NP_1 = P_1$ ($NP = P$), then NP_1 (NP) is closed under complement, and hence the class of (generalized) spectra is closed under complement. It is a famous open problem in automata theory as to whether $NP = P$; the evidence seems to be strongly against it. We remark that it is well known that $NP = P$ implies that $NP_1 = P_1$, and that if NP is closed under complement, then so is NP_1; these results follow, for example, by an obvious modification of an argument by Savitch [20, p. 186].

From Theorem 6, we see that spectra and generalized spectra are very broad classes. Most sets of positive integers that occur in number theory, such as the primes, the Fibonacci numbers, and the perfect numbers, are easily seen to be members of P_1, and a fortiori of NP_1. It is immediate from Theorem 6(2) that a set of positive integers is in NP_1 iff it is a spectrum.

THEOREM 7 (BENNETT [2]). *Assume that the set A of positive integers is in E^2_*. Then A and $\widetilde{A}$ are spectra.*

PROOF. By Theorem 3, A is recognizable by a linear-bounded automaton. So, by an easy, standard argument of counting the number of possible instantaneous descriptions, it follows that $A \in P_1$. So $A \in NP_1$, and hence A is a spectrum. Since E^2_* is closed under complement (Theorem 4), also $\widetilde{A} \in E^2_* \subseteq P_1 \subseteq NP_1$; hence $\widetilde{A}$ is a spectrum. □

It is an open problem as to whether there is any spectrum not in E^2_*.

Let BIN be the set of all spectra definable using only one graph predicate symbol. Obviously, if $S \in$ BIN, then S is definable using only one binary predicate symbol. The following result is proved in the author's doctoral dissertation [9].

THEOREM 8. *For each spectrum S, there is a positive integer k such that $\{n^k : n \in S\}$ is in BIN.*

We could not hope for it to be true that for each spectrum S, there is a positive integer k such that $\{n^k : n \in S\}$ is definable using only unary predicate symbols. This is because it is well known that by an elimination-of-quantifiers argument, it can be shown that each spectrum definable using only unary predicate symbols is either a finite or a cofinite set of positive integers.

We close this section by using Theorem 8 to show that if certain conjectures about spectra are false, then a counterexample occurs in BIN.

THEOREM 9. *The following two statements are equivalent.*

1. $NP_1 = P_1$.
2. $BIN \subseteq P_1$.

PROOF. $1 \Rightarrow 2$: BIN $\subseteq NP_1$, by Theorem 6(2).

$2 \Rightarrow 1$: Take S in NP_1; we want to show that $S \in P_1$. By the usual encoding arguments, we can assume that $S \subseteq Z^+$. By Theorem 8, we can find a positive integer k such that $T = \{n^k : n \in S\}$ is in BIN. Then $n \in S$ iff $n^k \in T$, for each positive integer n. So clearly, if $T \in P_1$, then $S \in P_1$. □

THEOREM 10. *The following two statements are equivalent.*

1. *If* $S \subseteq Z^+$, *then* $S \in NP_1$ *iff* $S \in E_2^*$.
2. $BIN \subseteq E_*^2$.

The proof is very similar to the previous proof. □

5. Categoricity. Call a first-order sentence *categorical* if it has at most one model (up to isomorphism) of each finite cardinality.

THEOREM 11. *Assume that the set* S *of positive integers is in* P_1. *Then there is a categorical sentence with spectrum* S.

PROOF. If the machine M in the proof of Theorem 6 is deterministic, then the sentence $\bigwedge\!\!\bigwedge_{i=1}^{9} \sigma_i$ defined in that proof is categorical. The "finite modification" which was called for to take care of small values of n is easily dealt with. □

So those naturally-occurring sets of positive integers that we discussed in the previous section are each the spectrum of a categorical sentence.

COROLLARY 12. *If* $NP_1 = P_1$, *then each spectrum is the spectrum of a categorical sentence.*

The proof is immediate. □

In the case of S-spectra, let us call a first-order $S \cup T$-sentence σ (where $S \cap T = \emptyset$) *S-categorical* if, whenever $\mathfrak{A}$ and $\mathfrak{B}$ are finite $S \cup T$-structures which are models of σ, and $\mathfrak{A} \upharpoonright S$ and $\mathfrak{B} \upharpoonright S$ are isomorphic, then so are $\mathfrak{A}$ and $\mathfrak{B}$.

If A is an S-spectrum, then it does not quite follow, as in Theorem 11, that if $E(A) \in P$ then there is T and there is an S-categorical $S \cup T$-sentence σ such that $A = \mathrm{Mod}_\omega \exists T\sigma$. For, there are many different ways to impose the linear ordering $<$. However, if structures had a "built-in" linear ordering, then we could surmount this difficulty. One approach is to consider only finite S-structures $\mathfrak{A}$ such that $|\mathfrak{A}| \subseteq Z^+$. We could let $<$ be a symbol which, like $=$, has an invariant interpretation; namely, if $a, b \in |\mathfrak{A}|$, where $|\mathfrak{A}| \subseteq Z^+$, then $a <^{\mathfrak{A}} b$ iff a is a smaller integer than b. Then the desired result mentioned above follows.

6. Possible Asser counterexamples. In § 1, we gave several simple examples of generalized spectra whose complements do not seem to be generalized spectra. These also serve as examples of NP sets which are probably not in P.

It is harder to find candidates for sets which are spectra but whose complements are not spectra, or which are in NP_1 but not in P_1. This is because, as we observed, most naturally-occurring sets of positive integers are in P_1, and hence, of course, so are their complements.

As we shall see in §9, there exists a spectrum S such that $\{n\colon 2^n \in S\}$ is not a spectrum. This gives us one class of possible counterexamples. In fact, Bennett [2] shows that $\{n\colon 2^n + 1$ is composite$\}$ is a spectrum, and asks whether $\{n\colon 2^n + 1$ is prime$\}$ is a spectrum. We will show that Bennett's result follows fairly simply from Theorem 6 (Bennett's proof is different). We will answer Bennett's question (affirmatively) by making use of a very surprising result by Vaughn Pratt (unpublished). We need the following simple theorem.

THEOREM 13. *Assume* $A \subseteq Z^+$. *If* $A \in NP$, *then* $\{n\colon 2^n + 1 \in A\} \in NP_1$.

PROOF. Assume that M is a nondeterministic Turing machine which recognizes A in polynomial time. We will define a nondeterministic Turing machine M' which recognizes $B = \{n\colon 2^n + 1 \in A\}$ in exponential time. Given input n, the machine M' prints the string that starts and ends with a 1 and has $(n-1)$ 0's in between. This is the number $2^n + 1$ in binary notation. Then M' simulates the action of M on input $2^n + 1$. It is easy to see that M' recognizes B nondeterministically in exponential time. □

It is simple to show that $C = \{n\colon n$ is composite$\}$ is in NP. For, let M be a nondeterministic Turing machine which, given input n, "guesses" a divisor m of n and then tests it; if m divides n, then M accepts n. Clearly M recognizes C nondeterministically in polynomial time. So, from Theorem 13, $\{n\colon 2^n + 1$ is composite$\}$ is in NP_1, and hence is a spectrum.

Pratt proved that $\{n\colon n$ is prime$\}$ is in NP. From this very interesting result, it follows immediately from Theorem 13 that $\{n\colon 2^n + 1$ is prime$\}$ is a spectrum.

For each set S of words, define S' to be $\{\mathrm{len}(n)\colon n \in S\}$. As candidates for sets in NP_1 which are not in P_1, Robert Solovay (personal communication) essentially suggested considering sets S', where $S \in P$. We will now show that in a certain sense, this class is sufficient for a counterexample. The proof gives an application to automata theory of the equivalence in Theorem 6.

THEOREM 14. *The following three statements are equivalent*:

1. $NP_1 = P_1$.
2. *If* $S \in P$, *then* $S' \in P_1$.
3. *If* $S \in NP$, *then* $S' \in P_1$.

PROOF. $3 \Rightarrow 2$: This is immediate, since $P \subseteq NP$.

$1 \Rightarrow 3$: Assume that $S \in NP$. Then $S' \in NP_1$. For, assume that M recognizes S nondeterministically in time $l \mapsto l^k$, for some k. We will construct a machine M' that recognizes S' nondeterministically in exponential time. Given input m, the machine M' first guesses a number n of length m.

Then M' simulates M on the input n. Clearly, M' recognizes S'; M' accepts m in S' in roughly m^k steps. So $S' \in NP_1 = P_1$.

$2 \Rightarrow 1$: Assume that $A \in NP_1$. By the usual encoding arguments, we can assume that $A \subseteq Z^+$ (if, instead, $A \subseteq \Sigma^*$ for the finite set Σ, then we find an encoding $t: \Sigma^* \longrightarrow Z^+$ for which there is a constant c such that $\text{len}(t(w)) = c \cdot \text{len}(w)$ for each w in Σ^*; the details are straightforward). By Theorem 6, we know that A is a spectrum. Assume for simplicity that $A = \{n: \langle n \rangle \models \exists Q\sigma\}$, where Q is a binary predicate symbol. The general case is similar. Let S be the following set:

$$\{m: \exists n(\text{len}(m) = n^2 + 1, \text{ and if the binary representation of } m \text{ is } 1^\frown b, \text{ and if } R \text{ is the binary relation on } n \text{ which is represented by } b \text{ in the obvious way, then } \langle n; R\rangle \models \sigma)\}.$$

We use $n^2 + 1$ instead of n^2 to allow for the possibility of b being a string of all 0's.

Then $S \in P$. For, as we saw in the proof of Theorem 6, there is a positive integer k and a deterministic machine M which can determine whether $\langle n; R\rangle \models \sigma$ within n^k steps for each n (and R); note that n^k is bounded by a fixed polynomial of the length of m.

Since $S \in P$, it follows by hypothesis that $S' \in P_1$. Now $n \in A$ iff $(n^2 + 1) \in S'$, for each positive integer n. So $A \in P_1$. □

In the next section, we will find several specific (generalized) spectra A ($\mathcal{A}$) such that the class of (generalized) spectra is closed under complement iff $\widetilde{A}$ ($\widetilde{\mathcal{A}}$) is a (generalized) spectrum.

7. Complete sets. We will now deal with the notions of reducibility and completeness, which are due to Cook [7] and Karp [16]. We will show that completeness implies a complement-completeness (Theorem 15(2)), and we will use this fact, along with Theorem 6(1) and results in Karp's paper [16], to find particular generalized spectra whose complements are generalized spectra iff the complement of every generalized spectrum is generalized spectrum. In particular, we will show that the class of directed graphs with a Hamilton cycle is such a "complete" generalized spectrum; we will also exhibit a monadic complete generalized spectrum. We will then find a complete spectrum, and will show that the existence of a complete spectrum implies the existence of a complete spectrum in BIN (which we can actually find).

Assume that Σ_1 and Σ_2 are finite sets, and $A \subseteq \Sigma_1^*$, $B \subseteq \Sigma_2^*$. *B is reducible* (*reducible*$_1$) *to* A, written $B \propto A$ ($B \propto_1 A$), if there is a function f: $\Sigma_2^* \longrightarrow \Sigma_1^*$ in Π (Π_1) such that, for each x in Σ_2^*, $x \in B$ iff $f(x) \in A$.

It is clear that $\propto$ and $\propto_1$ are transitive.

A set A is NP (NP_1)-*complete* if

1. $A \in NP$ (NP_1).
2. $B \propto A$ $(B \propto_1 A)$ for each B in $NP (NP_1)$.

THEOREM 15. *Let* A *be* $NP(NP_1)$-*complete. Then*

1. $NP = P$ $(NP_1 = P_1)$ *iff* $A \in P(P_1)$;
2. $NP(NP_1)$ *is closed under complement iff* $\widetilde{A} \in NP(NP_1)$.

PROOF. Assume that $B \subseteq \Sigma^*$, that $B \in NP(NP_1)$ and that $B \propto A$ $(B \propto_1 A)$. Find f in $\Pi(\Pi_1)$ such that $x \in B$ iff $f(x) \in A$, for each x in Σ^*. It is straightforward to check that if $A \in P(P_1)$, then $B \in P(P_1)$. Note that $x \in \widetilde{B}$ iff $f(x) \in \widetilde{A}$; hence, if $\widetilde{A} \in NP(NP_1)$, then $\widetilde{B} \in NP(NP_1)$. The other implications are obvious. □

Part 1 of Theorem 15 (in the $NP = P$ case) is due to Karp. Cook was the first to show that there exists an NP-complete set. This set is SAT, the set of encodings of satisfiable propositional formulas in "conjunctive normal form" $\bigwedge_i \bigvee_j A_{ij}$, where each A_{ij} is a propositional letter or its negation.

THEOREM 16 (COOK [7]). *SAT is* NP-*complete.*

In [16], SAT is shown to be reducible to certain other sets in NP, which are thus NP-complete. We now describe two such sets.

1. HAM is the set of encodings of $\{Q\}$-structures that have a Hamilton cycle, where Q is a binary predicate symbol.

2. HIT is the set of encodings of families of subsets of a set, for which there is a "hitting set." If the input is (the encoding of) a finite family $\{A_1, \cdots, A_n\}$, where each $A_i \subseteq \{s_1, \cdots, s_r\}$, then a hitting set is a set $W \subseteq \{s_1, \cdots, s_r\}$ such that $W \cap A_i$ contains exactly one element for each i.

THEOREM 17 (KARP, TARJAN, AND LAWLER [16]). *HAM and HIT are* NP-*complete.*

We can now demonstrate two particular generalized spectra whose complements are generalized spectra iff the complement of every generalized spectrum is a generalized spectrum. Let Q be a binary predicate symbol, and U a unary predicate symbol.

THEOREM 18. *Let* $A = \{\mathfrak{A} \in \mathrm{Fin}(Q)$: $\mathfrak{A}$ *has a Hamilton cycle*$\}$. *Then the class of generalized spectra is closed under complement iff the complement of the* $\{Q\}$-*spectrum* A *is a* $\{Q\}$-*spectrum.*

PROOF. $\Leftarrow$: This is immediate.

$\Leftarrow$: This would follow immediately from Theorems 6(1), 15(2) and 17 except for one technicality. Namely, if $\mathcal{B}$ is an S-spectrum, and if C is the complement of $\mathcal{B}$ in $\{0, 1, \#\}^*$, then C is not quite $E(\widetilde{\mathcal{B}})$, but instead is the union of $E(\widetilde{\mathcal{B}})$ and the set D of words in $\{0, 1, \#\}^*$ which are not the encoding of an S-structure. However, since D is easily (deterministic polynomial-time) recognizable, it is clear that $C \in NP$ iff $E(\widetilde{\mathcal{B}}) \in NP$, and so there is no problem. $\square$

It is very interesting that Theorem 18 is a statement of pure logic that seems on the surface to have nothing to do with automata theory. However, its proof is heavily dependent on automata theory.

THEOREM 19. *Let* $\mathcal{A} = \mathrm{Mod}_\omega \exists U \forall x \exists ! y(Qxy \wedge Uy)$. *Then the class of generalized spectra is closed under complement iff the complement of the* $\{Q\}$*-spectrum* $\mathcal{A}$ *is a* $\{Q\}$*-spectrum.*

PROOF. We will show that $\mathrm{HIT} \propto E(\mathcal{A})$. Since $E(\mathcal{A}) \in NP$ by Theorem 6(1), it follows that $E(\mathcal{A})$ is NP-complete. The proof can then be completed as in Theorem 18.

Assume that e is an encoding of the family $\{A_1, \cdots, A_n\}$ of certain subsets of $\{s_1, \cdots, s_r\}$. We can assume that $n \geq r$ by repeating the set A_n as often as necessary. Define a finite $\{Q\}$structure $\mathfrak{A}_e$ with $|\mathfrak{A}_e| = \{1, \cdots, n\}$ such that $\langle i, j\rangle \in Q^{\mathfrak{A}_e}$ iff $s_j \in A_i$, for each i and j. Let f be a function which (in general) maps e onto the encoding of $\mathfrak{A}_e$ (and which maps nonencodings onto a fixed nonencoding). It is straightforward to check that $e \in \mathrm{HIT}$ iff $f(e) \in E(\mathcal{A})$, and that $f \in \Pi$; therefore, $\mathrm{HIT} \propto E(\mathcal{A})$. $\square$

Note that $\mathcal{A}$ of Theorem 19 is a monadic $\{Q\}$-spectrum, that is, a $\{Q\}$-spectrum in which all of the "extra" predicate symbols are unary (in this case, there is only one extra predicate symbol, and it is unary). It is well known that if S is a set of unary predicate symbols, and $\mathcal{B}$ is a monadic S-spectrum (that is, all predicate symbols, "given" and "extra," are unary), then there is a first-order S-sentence σ such that $\mathcal{B} = \mathrm{Mod}_\omega \sigma$. Hence $E(\mathcal{B}) \in P$, as in the proof of Theorem 6. So Theorem 19 is a best possible result (short of resolving the generalized Asser problem). We remark that the author proved the following result about monadic generalized spectra in his doctoral dissertation [9].

THEOREM 20. *Let* $\mathcal{A}$ *be the class of nonconnected finite* $\{Q\}$*-structures, where* Q *is a binary predicate symbol (a finite* $\{Q\}$*-structure* $\langle A; R\rangle$ *is connected if, for each* a, b *in* A, *there is a finite sequence* $a_1, \cdots, a_n$ *of points in* A *such that* $a = a_1$, $b = a_n$, *and either* Ra_ia_{i+1} *or* $Ra_{i+1}a_i$, *for* $1 \leq i < n$). *Then* $\mathcal{A}$ *is a monadic* $\{Q\}$*-spectrum, but* $\widetilde{\mathcal{A}}$ *is not a monadic* $\{Q\}$*-spectrum. In particular, the class of monadic generalized spectra is not closed under complement.*

We will now produce a "universal" NP set and a "universal" NP_1 set. Each will be complete. The technique is similar to that of Book [4], who also shows the existence of an NP_1-complete set.

Some preliminary remarks are required. If M is a one-tape nondeterministic Turing machine that operates in time T, then it is easy to see that there is a constant c and a one-tape nondeterministic Turing machine M' that recognizes A_M in time cT, such that the range of the function δ for M' (which gives the table of transitions for M') contains only sets with at most two members (intuitively, M' has at most two options per move). Whenever there are two options then by some convention we label one the first option and the other the second option.

We momentarily restrict our attention to a subclass $\mathcal{M}$ of the class of those one-tape nondeterministic Turing machines that have at most two options per move, by making natural restrictions so that $\mathcal{M}$ will be countable: We require that the sets K (of states) and Γ (of tape symbols) be subsets of ω; it is also convenient to require that the set Σ of input symbols be $\{0, 1\}$, and that each machine in $\mathcal{M}$ recognize a set of (binary representations of) positive integers. We assign Gödel numbers to machines in the class $\mathcal{M}$ in such a way that a tape head can sweep through the encoding of the Gödel number to find out how to simulate the machine with that Gödel number on a given step. One such way involves essentially letting the Gödel number be the concatenation of the entries of the table of transitions, with the # sign used as a separator. Each tape symbol and state is encoded by a string in $\{0, 1\}^*$. For details, see [13, pp. 102–103]. Denote by T_i the machine with Gödel number i.

We now define a ternary relation V, which holds for certain triples $\langle i, s, n\rangle$ with i and n positive integers, and s in $\{0, 1\}^*$. For $V(i, s, n)$ to hold, it is first necessary for i to be the Gödel number of a machine T_i. Simulate the action of T_i on the input n, in the following way: If on the kth step in the simulation, there is an option, then take the first (second) option if the kth digit in s is a 0 (1); if s is not of length at least k, then halt and reject. Then $V(i, s, n)$ holds iff the number n is accepted in this simulation.

Let t be any standard one-one map from $(Z^+)^3$ onto Z^+ such that $t \in \Pi \cap \Pi_1$ and $t^{-1} \in \Pi \cap \Pi_1$, and such that each of a, b, and c is bounded by $t(a, b, c)$. We can now define two sets of positive integers which are "universal" in the usual sense with respect to NP (NP_1) sets.

$$\mathrm{UNIV} = \{t(i, a, n)\colon i, a, n \in Z^+ \text{ and } \exists s(\mathrm{len}(s) = \mathrm{len}(a) \text{ and } V(i, s, n))\},$$

$$\mathrm{UNIV}_1 = \{t(i, a, n)\colon i, a, n \in Z^+ \text{ and } \exists s(\mathrm{len}(s) = a \text{ and } V(i, s, n))\}.$$

THEOREM 21. (1) *UNIV is NP-complete.* (2) *$UNIV_1$ is NP_1-complete.*

PROOF. UNIV $\in NP$. For, we can define a multi-tape nondeterministic machine M which, given $t(i, a, n)$ as input, finds i, a, and n, guesses s in $\{0, 1\}^*$ such that $\operatorname{len}(s) = \operatorname{len}(a)$, and then does the obvious simulation. The point is that a is so large that the time of simulation is (except for bookkeeping) equal to the length of a, which is bounded by the length of $t(i, a, n)$; hence, UNIV $\in NP$. Similarly, $\text{UNIV}_1 \in NP_1$, since the time of simulation is roughly given by a, which is roughly 2^l, where l is the length of a.

Assume that $B \in NP$; we want to show that $B \propto$ UNIV. By the usual encoding arguments, we can assume that $B \subseteq Z^+$. Find i and k such that T_i recognizes B in time $l \longmapsto l^k$. Then $n \in B$ iff $t(i, a, n) \in$ UNIV, where a is a string of $(\operatorname{len}(n))^k$ tallies. Clearly the function $n \longmapsto t(i, a, n)$ is in Π. So UNIV is NP-complete.

Now assume that the set B of positive integers is in NP_1. Find i and k such that T_i recognizes B, and T_i accepts each n in B within n^k steps. Then $n \in B$ iff $t(i, n^k, n) \in \text{UNIV}_1$. So UNIV_1 is NP_1-complete. □

We are especially interested in part 2 of Theorem 21, which gives us a particular spectrum whose complement is a spectrum iff the complement of every spectrum is a spectrum. We record this in Theorem 22.

THEOREM 22. *The class of spectra is closed under complement iff the complement of the spectrum $UNIV_1$ is a spectrum.*

The proof is immediate. □

COROLLARY 23. *There is an NP_1-complete set in BIN. Thus, this is an example of a spectrum A in BIN such that $\widetilde{A}$ is a spectrum iff the complement of every spectrum is a spectrum.*

PROOF. Find k from Theorem 8 such that $A = \{n^k : n \in \text{UNIV}_1\}$ is in BIN. We remark that a simple analysis shows that in this case, k can be taken to be 5. Then $n \in \text{UNIV}_1$ iff $n^k \in A$, for each n. Hence $\text{UNIV}_1 \propto_1 A$, and so A is NP_1-complete. □

8. A Savitch-like result. Savitch [20] showed that any set that can be recognized nondeterministically in tape S can be recognized deterministically in tape S^2. If such a theorem were true for time bounds—for example, if there were a constant k such that any set that can be recognized nondeterministically in time T can be recognized deterministically in time T^k—then, of course, it would follow that $NP = P$ and $NP_1 = P_1$. It is quite surprising that this strong condition we are discussing is essentially equivalent to the apparently weaker condition that $NP = P$.

We will prove this in Theorem 24. Then we will generalize the result in various ways, and conclude by an analogy with Post's problem.

THEOREM 24. *The following two statements are equivalent:*

1. $NP = P$.

2. *There exists a constant k such that, for every countable function T with $T(l) \geqslant l + 1$ for each l and for every language A which is recognized by a nondeterministic one-tape Turing machine in time T, the language A is recognized by a deterministic one-tape Turing machine in time T^k.*

PROOF. $2 \Rightarrow 1$: This is immediate, since $l \longmapsto l^k$ is countable for each k.

$1 \Rightarrow 2$: It is sufficient to show that $1 \Rightarrow 2'$, where $2'$ is obtained from 2 by replacing both occurrences of "language A" by "set $A \subseteq Z^+$." This is because of simple interrelationships between machines M which recognize a language A and machines M' which recognize an encoding $A' \subseteq Z^+$ of A. The details are straightforward and nonunique, and are left to the reader.

Let $R = \{\bar{i} \# \bar{a} \# \bar{n}$: if $\bar{i}, \bar{a}$, and $\bar{n}$ are the binary representations of the positive integers i, a, and n, then $t(i, a, n) \in \mathrm{UNIV}\}$. Then $R \in NP$, and so by hypothesis (and by Theorem 2), there is a constant k' and a one-tape deterministic machine M_1 which recognizes R in time $l \longmapsto l^{k'}$. We can assume that $k' \geqslant 2$. Let $k = 2k'$.

Assume that A is recognized by a nondeterministic one-tape machine in time T, where T is countable and $T(l) \geqslant l + 1$ for each l. Then as we observed earlier, there is a constant c_1 and a machine T_{i_0} (with at most two options per move) which recognizes A in time $c_1 T$. Since T is countable, it is easy to see that $c_1 T$ is countable. Hence there is a constant c_2 and a deterministic two-tape machine M_2 which, for each l and each input w of length l on the first tape, prints at least $c_1 T(l)$ tallies on its second tape in at most $c_2 T(l)$ steps.

We will now describe a 3-tape nondeterministic machine M which recognizes A. Given input n of length l on its first tape, M simulates M_2 to print a string w of at least $c_1 T(l)$ tallies on its second tape in at most $c_2 T(l)$ steps. Then M prints $\bar{i}_0 \# w \# \bar{n}$ on its third tape in $\mathrm{len}(i_0) + \mathrm{len}(w) + l + 2$ steps. Now M simulates M_1 with $\bar{i}_0 \# w \# \bar{n}$ as input. This takes at most $(\mathrm{len}(i_0) + \mathrm{len}(w) + l + 2)^{k'}$ steps. Since $T(l) \geqslant l + 1$, since clearly $\mathrm{len}(w) \leqslant c_2 T(l)$, and since $\mathrm{len}(i_0) + 2$ is a constant, the total number of steps required is bounded by $((c_2 + 2)T(l))^{k'}$ for sufficiently large l. Clearly, M recognizes A. By Theorem 2, the set A is recognized by a one-tape deterministic machine in time $((c_2 + 2)T)^k$. Hence, by Theorem 1, A is recognized by a one-tape deterministic machine in time T^k. $\square$

By very similar proofs, we can demonstrate the following two results.

THEOREM 25. *The following two statements are equivalent:*

1. $NP_1 = P_1$.

2. *There exists a constant* k *such that, for every countable function* T *with* $T(l) \geqslant 2^l$ *for each* l *and for every language* A *which is recognized by a nondeterministic one-tape Turing machine in time* T, *the language* A *is recognized by a deterministic one-tape Turing machine in time* T^k.

THEOREM 26. *The following two statements are equivalent.*

1. NP (NP_1) *is closed under complement.*

2. *There exists a constant* k *such that, for every countable function* T *with* $T(l) \geqslant l + 1$ $(T(l) \geqslant 2^l)$ *for each* l *and for every language* A *which is recognized by a nondeterministic one-tape Turing machine in time* T, *the language* $\widetilde{A}$ *is recognized by a nondeterministic one-tape Turing machine in time* T^k.

We conclude this section by an analogy with Post's problem. Definitions and notation are from Rogers [19]. Post's problem asks whether there is an r.e. set C which is not Turing-equivalent to either $\emptyset$ or to the halting problem set K.

Let $\{W_x^B : x \in Z^+\}$ be an effective listing of all sets of natural numbers which are r.e. in B. As Rogers notes, if A and B are r.e., then the assertion that A is not Turing-reducible to B is equivalent to $\forall x(\widetilde{A} \neq W_x^B)$, or equivalently, $\forall x \exists y(y \in A$ iff $y \in W_x^B)$. If $(\exists$ recursive $f)(\forall x)(f(x) \in A$ iff $f(x) \in W_x^B)$, then we say that A *is constructively nonrecursive in* B.

Many attempts to solve Post's problem failed, because investigators tried to find some r.e. set C such that C is constructively nonrecursive in $\emptyset$ and such that K is constructively nonrecursive in C. Rogers shows that if A and B are r.e., and if A is constructively nonrecursive in B, then B is recursive. Hence, any such attempt must fail.

In an analogous way, one might wonder whether it is possible that $NP = P$, but that all attempts to prove this have failed because investigators have been searching for some recursive function f which maps the index i of each nondeterministic Turing machine into an index $f(i)$ of a deterministic machine which recognizes the same set, such that if the machine with index i operates in polynomial time, then so does the machine with index $f(i)$. We will now show that if $NP = P$, then there is such a recursive function f, as long as we restrict ourselves to machines that operate in a given polynomial time bound, such as machines that operate in time $l \longmapsto l^r$ for fixed r.

For each r, let T_i^r be a two-tape nondeterministic machine which, given input n on its first tape, simulates the action of T_i on n for at most $(\text{len}(n))^r$ steps, by using its second tape as a clock. If in the simulation T_i has not accepted within $(\text{len}(n))^r$ steps, then T_i^r halts and rejects.

THEOREM 27. *The following two statements are equivalent:*

1. $NP = P$.

2. *There exists a constant* k *and a function* f *in* Π *such that, for each Gödel number* i *and each positive integer* r, *the machine* $T_{f(i,r)}$ *is a one-tape deterministic Turing machine which operates in time* $l \longmapsto l^{kr}$, *and which recognizes the same set as* T_i^r. *Hence, if* T_i *operates (nondeterministically) in time* $l \longmapsto l^r$, *then* $T_{f(i,r)}$ *recognizes the same set as* T_i.

PROOF. This is clear from the proof of Theorem 24. □

9. A counterexample. We will show that there is a spectrum S in BIN such that $\{n: 2^n \in S\}$ is not a spectrum. By way of contrast, it is easy to see, because of Theorem 6(2), that for each spectrum S and each polynomial p with rational coefficients the set $\{n: p(n) \in S\}$ is a spectrum. The fact that there is a spectrum S such that $\{n: 2^n \in S\}$ is not a spectrum is extremely closely related, both in content and method, to Bennett's results on higher-order spectra [2], although he did not specifically state or prove this result. If we analyze Bennett's proof, then we see that he essentially proved that there is a spectrum S and a positive integer k such that $\{n: 2^{n^k} \in S\}$ is not a spectrum.

We will also show that our techniques give a new proof of a result of Book [3] that $NP \neq NP_1$.

LEMMA 28. *Let* A *be a spectrum. Then, for some constant* k, *the set* A *is recognized by a one-tape deterministic Turing machine in tape* $l \longmapsto 2^{kl}$.

PROOF. Assume first that $A = \{n: \langle n \rangle \models \exists Q\sigma\}$, where Q is a binary predicate symbol. Define a one-tape deterministic machine M which, given input n, systematically prints all possible strings in $\{0, 1\}^*$ of length n^2, and tests them one by one to see if the binary relation R on n which the string represents in the natural way has the property that $\langle n; R \rangle \models \sigma$. M accepts n iff it finds some such string. If $\mathrm{len}(n) = l$, then $n^2 < 2^{2l}$. Hence M can be arranged to operate in tape $l \longmapsto 2^{3l}$. Similarly, for each spectrum S there is a constant k such that A is recognizable in tape $l \longmapsto 2^{kl}$. □

LEMMA 29. *There is a set* $A \subseteq Z^+$ *which is recognized by a one-tape deterministic Turing machine in tape* $l \longmapsto 2^{l^2}$, *which is not a spectrum.*

PROOF. This follows from Theorem 5 and Lemma 28, since it is easy to see that $l \longmapsto 2^{l^2}$ is constructible and that $\liminf_{l \to \infty} 2^{kl}/2^{l^2} = 0$ for each k.

LEMMA 30. *Assume that* $A \subseteq Z^+$ *is recognized by a one-tape deterministic Turing machine in tape* $l \longmapsto 2^{l^2}$. *Then there is a set* B *in* E_*^2 *such that* $A = \{n: 2^{2^n} \in B\}$.

PROOF. Let M be a one-tape deterministic Turing machine that recognizes A in tape $l \mapsto 2^{l^2}$. Let M_1 be a one-tape deterministic Turing machine which operates as follows: Given input m, the machine M_1 tests to see if there is a positive integer n such that $m = 2^{2^n}$. If not, then M_1 rejects. If so, then M_1 simulates M on input n. Now M_1 can be designed to be a linear-bounded automaton. This is because $\mathrm{len}(2^{2^n}) = 2^n + 1$, which is bigger than $2^{2^{l-1}}$ (where $l = \mathrm{len}(n)$), which is bigger than 2^{l^2} for sufficiently large l. So, by Theorem 3, the set B which M_1 recognizes is in $\mathcal{E}^2_*$. Clearly $A = \{n: 2^{2^n} \in B\}$. $\square$

THEOREM 31. *There is a spectrum* S *such that* $\{n: 2^n \in S\}$ *is not a spectrum.*

PROOF. Find A from Lemma 29 and B from Lemma 30 such that A is not a spectrum, $B \in \mathcal{E}^2_*$, and $A = \{n: 2^{2^n} \in B\}$. By Theorem 7, we know that B is a spectrum. Let $C = \{n: 2^n \in B\}$. Then $A = \{n: 2^n \in C\}$. Assume that it is always true that whenever S is a spectrum, then $\{n: 2^n \in S\}$ is a spectrum. Then C is a spectrum (since B is), and so A is a spectrum (since C is). But this is a contradiction. $\square$

COROLLARY 32. *There is a spectrum* T *in* BIN *such that* $\{n: 2^n \in T\}$ *is not a spectrum.*

PROOF. Find S from Theorem 31 such that $D = \{n: 2^n \in S\}$ is not a spectrum. Find a positive integer k from Theorem 8 such that $T = \{n^k: n \in S\}$ is in BIN. Let $E = \{n: 2^n \in T\}$. Then $n \in D$ iff $kn \in E$, for each positive integer n; for, $n \in D$ iff $2^n \in S$ iff $2^{kn} \in T$ iff $kn \in E$. If E were a spectrum, then E would be in NP_1, and so clearly D would be in NP_1. Hence D would be a spectrum, a contradiction. $\square$

We close this section with some further observations. Theorem 13 of §6 could just as well have been stated as follows:

(4) *Assume* $A \subseteq Z^+$. *If* $A \in NP$, *then* $\{n: 2^n \in A\}$ *is in* NP_1.

We remark that we can use the technique of the proof of Lemma 30 to show that (4) has a converse:

THEOREM 33. *Assume* $B \subseteq Z^+$. *Then* $B \in NP_1$ *iff there is* A *in* NP *such that* $B = \{n: 2^n \in A\}$.

Because of Theorem 6(2), we know that Theorem 31 can be restated as follows:

(5) *There is a set* A *in* NP_1 *of positive integers such that* $\{n: 2^n \in A\}$ *is not in* NP_1.

Similarly, we can prove the following:

THEOREM 34. *There is a set A in NP of positive integers such that $\{n: 2^n \in A\}$ is not in NP.*

Finally, we observe that (4) and Theorem 34 combine to give us a theorem of Book:

THEOREM 35 (BOOK [3]). $NP \subsetneq NP_1$.

PROOF. From Theorem 34, find a set A in NP of positive integers such that $B = \{n: 2^n \in A\}$ is not in NP. By (4), we know that $B \in NP_1$. So $NP \neq NP_1$. Of course, $NP \subseteq NP_1$. □

Book's proof depends on a fairly difficult result of Cook [8]. No simple diagonalization argument seems capable of proving Theorem 35 directly, because we are dealing with nondeterministic, rather than deterministic, time-complexity classes. However, a simple diagonalization argument does show that $P \subsetneq P_1$.

10. A real-time recognizable NP-complete set. We conclude by exhibiting an NP-complete set REAL which is recognized by a nondeterministic two-tape machine in real time. The existence of such a set is not new: Hunt [14] shows the existence of an NP-complete set which is recognizable nondeterministically in linear time, and Book and Greibach [5] prove that every set recognizable nondeterministically in linear time is recognizable by a nondeterministic two-tape Turing machine in real time. However, our set is produced directly, and is fairly simple. The existence of such a set is a best-possible result, since Rabin and Scott [17] show that every set which is recognized by a one-tape nondeterministic Turing machine in real time is recognized by a one-tape deterministic Turing machine in real time.

Let REAL $= \{a_1 \# a_2 \# \cdots \# a_{2r}: r \in Z^+; a_i \in \{0, 1, 2\}^*$ for each i; $\text{len}(a_i) = \text{len}(a_j)$ for each i, j; and there exists b in $\{0, 1\}^*$ such that $\text{len}(b) = \text{len}(a_i)$ for each i, and such that for each odd i there exists k such that the kth member of the string b and the kth member of the string a_i are the same$\}$.

THEOREM 36. *REAL is an NP-complete set which is recognized by a two-tape nondeterministic Turing machine in real time.*

PROOF. Let M be a two-tape nondeterministic Turing machine which works as follows: As a_1 is being read on the first, or input tape, M nondeterministically prints some b in $\{0, 1\}^*$ on the second tape, such that $\text{len}(b) = \text{len}(a_1)$; meanwhile, M checks to make sure that, for some k, the kth digit of b

is the same as the kth digit of a_1. When M reads # on the input tape and starts reading a_2, the second tape head runs back over b on the second tape and uses the length of b to measure the length of a_2. If len$(a_2) \neq$ len(b), then M halts and rejects. If len$(a_2) =$ len(b), then the tape heads are in a position to compare b and a_3 digit by digit. M continues in the obvious way. Clearly, M recognizes REAL in real time.

SAT $\propto$ REAL: Let θ be a conjunctive normal form expression, with clauses $C_1, \cdots, C_r$, and propositional letters $A_1, \cdots, A_n$. (If $\theta = \bigwedge_i \bigvee_j B_{ij}$, then each $\bigvee_j B_{ij}$ is a clause.) We can assume that no clause C_i contains both A_k and $\sim A_k$ for any k, or else that clause can be eliminated. Let β_θ be the expression $a_1 \# a_2 \# \cdots \# a_{2r}$, where each a_i is of length n, where if i is even, then a_i is a string of tallies, and where if $i = 2s - 1$ is odd, then for each k $(1 \leqslant k \leqslant n)$, the kth digit of a_i is as follows:

$$\begin{cases} 0, & \text{if } \sim A_k \text{ appears in the } s\text{th clause,} \\ 1, & \text{if } A_k \text{ appears in the } s\text{th clause,} \\ 2, & \text{otherwise.} \end{cases}$$

For any reasonable encoding e, there exists a constant c such that if the encoding $e(\theta)$ of θ is of length l, then $l \geqslant c \cdot \max(r, n)$. Now β_θ has length $2rn + 2r - 1$, which is dominated by $2l^2/c + 2l/c - 1$. So if f is the function which (in general) maps $e(\theta)$ onto β_θ (and which maps strings not of the form $e(\theta)$ onto a fixed string not in REAL), then it is easy to see that $f \in \Pi$ (we are assuming that $\{e(\theta): \theta$ is a formula in conjunctive normal form$\}$ is in P, which is also true for any reasonable encoding e). Most importantly, it is clear that θ is satisfiable iff $\beta_\theta \in$ REAL. Hence, SAT $\propto$ REAL. □

Bibliography

1. G. Asser, *Das Repräsentantenproblem im Prädikatenkalkül der ersten Stufe mit Identität*, Z. Math. Logik Grundlagen Math. 1 (1955), 252–263. MR 17, 1038.

2. J. H. Bennett, *On spectra*, Doctoral Dissertation, Princeton University, Princeton, N. J., 1962.

3. R. V. Book, *On languages accepted in polynomial time*, SIAM J. Comput. 1 (1972), 281–287.

4. ———, *Comparing complexity classes* (submitted for publication).

5. R. V. Book and S. Greibach, *Quasi-realtime languages*, Math. Systems Theory 4 (1970), 97–111. MR 43 #1772.

6. A. Cobham, *The intrinsic computational difficulty of functions*, Logic, Methodology, and Philos. (Proc. 1964 Internat. Congress), North-Holland, Amsterdam, 1965, pp. 24–30. MR 34 #7376.

7. S. Cook, *The complexity of theorem-proving procedures*, Conference Record of Third ACM Sympos. on Theory of Computing, 1970, pp. 151–158.

8. S. Cook, *A hierarchy for nondeterministic time complexity*, Proc. Fourth ACM Sympos. on Theory of Computing, 1972, pp. 187–192.

9. R. Fagin, *Contributions to the model theory of finite structures*, Doctoral Dissertation, University of California, Berkeley, Calif., 1973.

10. A. Grzegorczyk, *Some classes of recursive functions*, Rozprawy Mat. 4 (1953), pp. 1–45. MR 15, 667.

11. J. Hartmanis, P. M. Lewis II and R. E. Stearns, *Hierarchies of memory limited computations*, IEEE Conference Record on Switching Circuit Theory and Logical Design, Ann Arbor, Mich., 1965, pp. 179–190.

12. J. Hartmanis and R. E. Stearns, *On the computational complexity of algorithms*, Trans. Amer. Math. Soc. 117 (1965), 285–306. MR 30 #1040.

13. J. E. Hopcroft and J. D. Ullman, *Formal languages and their relation to automata*, Addison-Wesley, Reading, Mass., 1969. MR 38 #5533.

14. H. B. Hunt III, *On the time and tape complexity of languages*, Technical Report 73–156, Cornell Univ., Ithaca, N. Y., Jan. 1973.

15. N. D. Jones and A. L. Selman, *Turing machines and the spectra of first-order formulas with equality*, Proc. Fourth ACM Sympos. on Theory of Computing, 1972, pp. 157–167.

16. R. M. Karp, *Reducibility among combinatorial problems*, Technical Report 3, University of California, Berkeley, April 1972; Also in *Complexity of computer computations* (ed. R. E. Miller et al.), Plenum Press, New York, 1972.

17. M. O. Rabin and D. Scott, *Finite automata and their decision problems*, IBM J. Res. Develop. 3 (1959), pp. 114–125; Also in *Sequential machines: selected papers* (ed. E. F. Moore), Addison-Wesley, Reading, Mass., 1964, pp. 63–91. MR 21 # 2559.

18. R. W. Ritchie, *Classes of predictably computable functions*, Trans. Amer. Math. Soc. 106 (1963), 139–173. MR 28 #2045.

19. H. Rogers, *Theory of recursive functions and effective computability*, McGraw-Hill, New York, 1967. MR 37 #61.

20. W. J. Savitch, *Relationships between nondeterministic and deterministic tape complexities*, J. Comput. Systems Sci. 4 (1970), 177–192. MR 42 #1605.

21. H. Scholz, J. Symbolic Logic 17 (1952), 160.

22. J. R. Shoenfield, *Mathematical logic*, Addison-Wesley, Reading, Mass., 1967. MR 37 #1224.

23. A. Tarski, *Contributions to the theory of mdoels.* I, II, Nederl. Akad. Wetensch. Proc. Ser. A. 57 = Indag. Math. 16 (1954), 572–588. MR 16, 554.

24. A. M. Turing, *On computable numbers with an application to the Entscheidungsproblem*, Proc. London Math Soc. (2) 42 (1936), 230–265; correction, ibid. (2) 43 (1937) 544–546.

T. J. WATSON RESEARCH CENTER, IBM