THE JOURNAL OF SYMBOLIC LOGIC
Volume 69, Number 1, March 2004

P VERSUS NP AND COMPUTABILITY THEORETIC CONSTRUCTIONS IN COMPLEXITY THEORY OVER ALGEBRAIC STRUCTURES

GUNTHER MAINHARDT

Abstract. We show that there is a structure of countably infinite signature with $P = N_2P$ and a structure of finite signature with $P = N_1P$ and $N_1P \neq N_2P$. We give a further example of a structure of finite signature with $P \neq N_1P$ and $N_1P \neq N_2P$. Together with a result from [10] this implies that for each possibility of P versus NP over structures there is an example of countably infinite signature. Then we show that for some finite $\mathscr{L}$ the class of $\mathscr{L}$-structures with $P = N_1P$ is not closed under ultraproducts and obtain as corollaries that this class is not Δ-elementary and that the class of $\mathscr{L}$-structures with $P \neq N_1P$ is not elementary. Finally we prove that for all f dominating all polynomials there is a structure of finite signature with the following properties: $P \neq N_1P$, $N_1P \neq N_2P$, the levels $N_2TIME(n^i)$ of N_2P and the levels $N_1TIME(n^i)$ of N_1P are different for different i, indeed $DTIME(n^{i'}) \not\subseteq N_2TIME(n^i)$ if $i' > i$; $DTIME(f) \not\subseteq N_2P$, and $N_2P \not\subseteq DEC$. DEC is the class of recognizable sets with recognizable complements. So this is an example where the internal structure of N_2P is analyzed in a more detailed way. In our proofs we use methods in the style of classical computability theory to construct structures except for one use of ultraproducts.

§1. Introduction and basic concepts. An important problem of classical complexity theory is the question, whether $P = NP$ holds. Though this problem is still open, it has been solved relative to oracles; in [1] it was proved that oracles with $P = NP$ as well as oracles with $P \neq NP$ exist. Another way to relativize this problem is to consider it relative to structures in the sense of mathematical logic rather than oracles, such that we can make use of the relations, functions, and constants of the structure. Over structures, described below, we have two kinds of nondeterminism: A first kind, which corresponds to the situation that we have finitely many choices for the next computation step and a second kind, which consists of guessing an element of the structure. Concerning the second kind, note that there are structures with infinitely many elements and that it allows guessing solutions of problems in "NP". Hence, over structures we have two versions of NP, N_1P, where only the first kind is allowed, and N_2P, where additionally the second kind is allowed. This trivially yields $P \subseteq N_1P \subseteq N_2P$ over any structure. Similar remarks hold for other questions of classical complexity theory, too.

Computability theory over structures gives a possibility to generalize algorithms, also to other domains than the natural numbers or words over a finite alphabet.

Section 2 is concerned with P versus NP over structures.

Of course, structures can also be considered from the point of view of model theory and we should address the question which connections there are between model

Received February 10, 2002; revised March 4, 2003.

0022-4812/04/6901-0005/$3.60

theory and computability theory. In Section 3 about $P = N_1P$ and ultraproducts we give an example of such a connection suggested by the proof of a theorem of Section 2.

In the final Section 4 we prove our result concerning the more detailed analysis of N_2P.

The reason why we give computability theoretic constructions of structures is first of all that up to now some results can only be proved using such constructions, e.g., Theorems 1, 2, 4 and Corollary 1. These methods are quite flexible, various results can be proved in computability theory over structures, some of which are presented in this paper and they provide a further way to construct structures. In some cases it is possible to combine different strategies in one construction, so that it is possible to obtain structures having several properties at the same time, see Theorem 5. Since structures also have algebraic and model theoretic aspects, we think that it is an interesting question how far we get using methods from classical computability theory like diagonalizations in computability theory over structures. Our proof of Theorem 3 does not yield a natural example of a structure but is technically simpler and more direct than another proof giving a natural example. Our approach makes it possible to bring together ideas from complexity theory, recursion theory, and model theory as will be demonstrated in this paper.

We next define the basic concepts of this paper, structures, our model of computation over structures, and the corresponding complexity classes. The model is essentially that of [6], [7], and [8]. For more on structures see textbooks on mathematical logic, e.g., [5], and more on our model of computation, its elementary properties and the complexity classes can be found in [6] and [7], concerning infinite signature see also [8]. In these papers the model was first introduced. So our paper is based on these papers but can be read independently. The reader should be familiar with basic complexity theory, a reference is [2] and with the basics of classical recursion theory, which can be found in [13]. For model theory see [3] or [4].

First, we are concerned with structures, where intuitively, a structure is a nonempty set A together with some functions and relations built from A and some distinguished elements of A, called constants. A *language* $\mathscr{L}$ in the sense of mathematical logic is a collection of n-ary function symbols, m-ary relation symbols, and constant symbols, where the unique arities $n \geq 1$ and $m \geq 1$ depend on the symbols. The sets of function symbols, of relation symbols, and of constant symbols are pairwise disjoint. An *$\mathscr{L}$-structure $\mathscr{A}$* is a pair $(A, \mathscr{I})$, where A is a nonempty set and $\mathscr{I}$ is a mapping with domain $\mathscr{L}$ which assigns to each n-ary function symbol F a function $F^{\mathscr{A}}: A^n \to A$, to each m-ary relation symbol R a relation $R^{\mathscr{A}} \subseteq A^m$ and to each constant symbol c a constant $c^{\mathscr{A}} \in A$. We say $\mathscr{I}$ *interprets* the symbols in $\mathscr{L}$. A *structure* is an $\mathscr{L}$-structure for some $\mathscr{L}$. For our purposes, it suffices to define the *signature* of a structure as the cardinality of the corresponding $\mathscr{L}$, which can be a natural number. By saying e.g., that a structure $(A, \mathscr{I})$ contains a certain function, we mean that this function is in the range of $\mathscr{I}$. In defining structures we are often not precise concerning the domain of $\mathscr{I}$, we will always only define the universe and the range of $\mathscr{I}$ precisely. For a structure $(A, \mathscr{I})$ the set A is called its *universe*.

For any nonempty set A we use the following notation: A^+ is the set of all nonempty finite sequences over A. For $w \in A^+$, $|w|$ denotes the length of w.

Finite sequences are also called *strings*. We write nonempty strings in the form $(a_1, a_2, \dots, a_n)$ instead of $a_1 a_2 \dots a_n$ which is advantageous e.g., if $A = \mathbf{N}$ and if we use the binary representation. By S we denote the successor function on $\mathbf{N}$. For the string $w \in \mathbf{N}^+$ consisting of k zeroes we write 0^k. We use $\subset$ to denote proper inclusion.

Now we turn to the model of computation. Our idea for performing computations over a structure $(A, \mathscr{I})$ is to have a finite tape, the input and work tape, with the tape cells containing elements of A and to have several pointers each one scanning such a cell. Then we can perform computations similar to the case of classical Turing machines, where here we can make use of the functions, relations, and constants of $(A, \mathscr{I})$ and we can guess elements of A. The time complexity will be measured in the length of the input, which is a finite sequence consisting of elements of A. We remark, that our proofs should work over any reasonable model of computation over structures.

We now define $\mathscr{L}$-programs for a language $\mathscr{L}$ as syntactical objects. All symbols newly introduced thereby are assumed to be different from elements of $\mathscr{L}$. $\mathscr{L}$-programs will be used for computations over $\mathscr{L}$-structures as described below. Let p_j, $j \in \mathbf{N}$, be a sequence of distinct symbols, called *pointer variables*. We have three types of *atomic pointer expressions*, namely for $j, j' \geq 0$

$$p_j \equiv p_{j'},$$
$$r\text{-}end(p_j),$$
$$l\text{-}end(p_j).$$

For each pointer variable p_j there is a *data variable* $\tilde{p}_j$. *Basic data terms* are defined to be either data variables, constant symbols from $\mathscr{L}$ or have the form $F(\tilde{p}_{j_1}, \dots, \tilde{p}_{j_l})$ for an l-ary function symbol F from $\mathscr{L}$.

Basic data expressions are of the form $\tilde{p}_j \equiv \tilde{p}_{j'}$ or of the form $V(\tilde{p}_{j_1}, \dots, \tilde{p}_{j_m})$ for an m-ary relation symbol V from $\mathscr{L}$. We have the following *unconditional instructions*:

(i) assignment instructions: $\tilde{p}_j := t$ with a basic data term t,
(ii) pointer move instructions: $r\text{-}move(p_j)$ or $l\text{-}move(p_j)$,
(iii) append instructions: $r\text{-}app(p_j)$ or $l\text{-}app(p_j)$,
(iv) delete instructions: $del(p_j)$,
(v) stop instructions: *halt*, which can be accepting,
(vi) jump instructions: $goto(m_0, \dots, m_n)$ with $n, m_0, \dots, m_n \in \mathbf{N}$,
(vii) guess instructions: $guess(\tilde{p}_j)$.

A *conditional instruction* has the form

if Cond then Inst

with an unconditional instruction *Inst* and an atomic pointer or basic data expression *Cond*.

A *program*, more precisely $\mathscr{L}$*-program*, N is a finite sequence

$$(B_0, \dots, B_l),\ l \in \mathbf{N},$$

of conditional or unconditional instructions B_i, $0 \leq i \leq l$. For convenience we require that B_l is a stop instruction. Below, if N is a program with input w, we also write $N(w)$ for short. Programs, in which instructions of the form (vii) do not appear, and where always $n = 0$ in instructions of the form (vi) are called *deterministic.* Programs, in which instructions of the form (vii) do not appear, are called *nondeterministic of the first kind.* Arbitrary programs are also called *nondeterministic of the second kind.*

We next define the notion of a configuration of an $\mathscr{L}$-program. For our purposes it is useful to give a very abstract definition. So let an $\mathscr{L}$-program $N = (B_0, \ldots, B_l)$ be given and $k = max\{j\colon p_j \text{ or } \tilde{p}_j \text{ appears in } N\}$, $(max\,\emptyset = 0)$. Then a *configuration* of N is a tuple

$$(z; m; n_0, \ldots, n_k),$$

where $z = (a_0, \ldots, a_i)$ is an arbitrary nonempty finite sequence, $0 \le m \le l$, and for $0 \le j \le k$ we have $0 \le n_j \le i$. The idea behind this definition is that z is the string currently processed by N, m is the index of the next instruction to be executed, and that for $0 \le j \le k$ we have a pointer that points to position n_j in z. Moreover, for $0 \le j \le k$ we associate the pointer position n_j of the jth pointer with the pointer variable p_j and a_{n_j}, which corresponds to this position, with the data variable $\tilde{p}_j$. The reason why z has to be nonempty is that we always want the pointers to point to some position. Moreover, if B_m is an accepting stop instruction, then we say that the configuration is *accepting*.

In the following, let an $\mathscr{L}$-structure $\mathscr{A}$ with universe A be given. We want to introduce the notion of a computation of an $\mathscr{L}$-program $N = (B_0, \ldots, B_l)$ on a string $w \in A^+$. Therefore, we only consider configurations of N with $z \in A^+$. Again $k = max\{j\colon p_j \text{ or } \tilde{p}_j \text{ appears in } N\}$. We first define, in an informal way, how *successor configurations* of a certain configuration of N are obtained. So let a configuration

$$d = (z; m; n_0, \ldots, n_k),\ z = (a_0, \ldots, a_i),\ z \in A^+,$$

of N be given. We associate with each basic data term t, containing at most the data variables $\tilde{p}_0, \ldots, \tilde{p}_k$ an element $t^d \in A$. If $t = c$, c a constant symbol, then $t^d = c^{\mathscr{A}}$, if $t = \tilde{p}_j$, $0 \le j \le k$, then $t^d = a_{n_j}$, (element in pointer position), and if $t = F(\tilde{p}_{j_1}, \ldots, \tilde{p}_{j_l})$, then $t^d = F^{\mathscr{A}}(a_{n_{j_1}}, \ldots, a_{n_{j_l}})$, (elements in pointer positions). Further, with each basic data expression $\tilde{p}_j \equiv \tilde{p}_{j'}$ with $0 \le j, j' \le k$, we associate the value 1 if $a_{n_j} = a_{n_{j'}}$ and 0 otherwise. Similarly, with each basic data expression $V(\tilde{p}_{j_1}, \ldots, \tilde{p}_{j_m})$ with $0 \le j_1, \ldots, j_m \le k$, we associate the value 1 if $(a_{n_{j_1}}, \ldots, a_{n_{j_m}}) \in V^{\mathscr{A}}$ and 0 otherwise. With each atomic pointer expression $p_j \equiv p_{j'}$, *l-end*(p_j), *r-end*(p_j), $0 \le j, j' \le k$, we associate the value 1 if $n_j = n_{j'}$, (same pointer position), $n_j = 0$, (left end of z), $n_j = i$, (right end of z), and 0 otherwise, respectively. Intuitively, 1 for true, 0 for false.

If B_m is an instruction (i), $\tilde{p}_j := t$, then the unique successor configuration of d is obtained by replacing a_{n_j}, the element in the position of the jth pointer, by t^d, (t^d clearly defined), and by replacing m by $m + 1$. If B_m is of the form (ii) and the jth pointer does not point to the right/left end of z, then we move it one position to the right/left, otherwise no move is performed. We replace m by $m + 1$. If B_m is of the form (iii) and if the jth pointer does not point to the right/left end of z, only m is replaced by $m + 1$; otherwise we enlarge z by appending a_i/a_0 to the right/left and we place the jth pointer on the new right/left end, the other pointers do not move, (so in the "left"-case the corresponding $n_{j'}$ change) and again m is replaced by $m+1$. If B_m is of the form (iv) and if the length of z is < 2 or the jth pointer does not point to an end of z, then we only replace m by $m+1$; otherwise we delete the element in the position of the jth pointer and move all pointers in that position to the new end position, finally we replace m by $m+1$. If B_m is of the form (v), then there is no successor configuration. If B_m is of the form (vi), then exactly for each $m_j \leq l$ there is one successor configuration, which is obtained by replacing m by m_j. If B_m is of the form (vii), then for each $a \in A$ we have a successor configuration obtained by replacing a_{n_j} by a and m by $m+1$. Finally, if B_m is a conditional instruction, then with *Cond* we have associated either 1 or 0. In case of 1 we proceed as if B_m was *Inst* and otherwise we replace m by $m+1$.

Then a *computation* of an $\mathscr{L}$-program N on a string $w \in A^+$ is a finite or infinite sequence of configurations of N such that: The first configuration in the sequence is $(w; 0; 0, \dots, 0)$, where all pointers point to the left end of w; all other configurations are successor configurations of their previous ones; if the sequence is finite, then the last configuration is a configuration without successor configuration. If such a computation ends in an accepting configuration, then we say, that the computation is *accepting* and that N *accepts* w.

We give an example. Suppose $\mathscr{L} = \{R\}$, R a binary relation symbol and $\mathscr{A} = (A, R^{\mathscr{A}})$ is an $\mathscr{L}$-structure. Then the following program N accepts an input $(a_0, \dots, a_n)$ iff there is a b with $bR^{\mathscr{A}}a_0, \dots, bR^{\mathscr{A}}a_n$. $N = (B_0, \dots, B_7)$, where $B_0 = \textit{l-app}(p_0)$, $B_1 = \textit{guess}(\tilde{p}_0)$, $B_2 = \textit{if } R(\tilde{p}_0, \tilde{p}_1) \textit{ then goto}(4)$, $B_3 = \textit{halt}$, $B_4 = \textit{if r-end}(p_1) \textit{ then halt}$, $B_5 = \textit{r-move}(p_1)$, $B_6 = \textit{goto}(2)$, $B_7 = \textit{halt}$. B_3 and B_7 are not accepting but the stop instruction in B_4 is accepting.

Now we are ready to define complexity classes over an $\mathscr{L}$-structure $\mathscr{A}$ with universe A. The set *recognized* by an $\mathscr{L}$-program N over $\mathscr{A}$ is

$$L(N) = \{w \in A^+ : N \text{ accepts } w \text{ over } \mathscr{A}\}.$$

The complexity class P consists of all $L(N)$ with N deterministic, such that for some polynomial p and all $w \in L(N)$, there is an accepting computation of $N(w)$ of length $\leq p(|w|)$. N_1P and N_2P are defined similarly, using programs N nondeterministic of the first and second kind, respectively. Note that, if N is deterministic, then for each $w \in A^+$ there is exactly one computation of N on w over $\mathscr{A}$. Analogous definitions hold for other e.g., linear time bounds and can be made for space bounds.

In this paper we are also interested in the following classes: Given $f : \mathbf{N} \to \mathbf{N}$, $\mathit{DTIME}(f)$ consists of all $L(N)$, N deterministic, such that for all $w \in L(N)$ there is an accepting computation of $N(w)$ of length $\leq O(f(|w|))$. It is clear how to define $N_1\mathit{TIME}(f)$ and $N_2\mathit{TIME}(f)$.

DEC consists of all $L \subseteq A^+$ such that as well L as $A^+ \setminus L$ are recognized by some program nondeterministic of the second kind.

We say that a function $f : A^+ \to A^+$ is *computable in linear time over* $\mathscr{A}$ if there is a deterministic program N, such that for all $w \in A^+$ the computation of N on w over $\mathscr{A}$ has length $\leq O(|w|)$ and ends in a configuration of the form $(f(w); m; 0, \dots, 0)$.

We call a function $f : \mathbf{N} \to \mathbf{N}$ *time constructible* iff for all $\mathscr{L}$ there is a deterministic $\mathscr{L}$-program M such that over all $\mathscr{L}$-structures, M generates a computation of length $g(n)$ on an input of length n for some $g \in O(f)$ and halts in a configuration $(z; m; 0, \ldots, 0)$ with $|z| = f(n)$.

We say $f\colon \mathbf{N} \to \mathbf{N}$ dominates $h\colon \mathbf{N} \to \mathbf{N}$ iff $f(n) \geq h(n)$ for all but finitely many n.

Remarks: It is easy to see, that over the structure $(\{0, 1\}, 0, 1)$ we have $N_1P = N_2P$ and the P versus N_1P question is equivalent to the classical P-NP-problem.

We obtain the same classes P, N_1P and N_2P if in the above definition we additionally require that all computations of $N(w)$ are of length $\leq p(|w|)$, analogously for other time bounds; so over $(A, \mathcal{I})$, $L \in P$ iff $A^+ \setminus L \in P$. These facts will be immediate by Lemma 2, its proof and the remark following Lemma 2.

This model of computation slightly differs from that of [6], [7], and [8]. But both concepts are essentially equivalent, e.g., yield the same classes P, N_1P, N_2P, because we can explicitly compute the values of the data terms of the other model in our model.

We give some examples. Over $(\mathbf{N}, 2^n)$ the function $f\colon \mathbf{N}^+ \to \mathbf{N}^+$ with $f(a_0, \ldots, a_n) = (2^{a_0}, \ldots, 2^{a_n})$ is computable in linear time. Over $(\mathbf{N}, +)$ the set of tuples $(a_0, \ldots, a_n)$, such that all a_i are even, is in N_2P. If R partially orders A, then the set of all $(a_0, \ldots, a_n)$ with $a_0Ra_1, a_1Ra_2, \ldots, a_{n-1}Ra_n$ is in P over (A, R).

§2. P versus NP over structures. As we have already mentioned, $P \subseteq N_1P \subseteq N_2P$ over any structure. Moreover, over any structure exactly one of the four following possibilities holds:

(i) $P = N_1P = N_2P$,
(ii) $P = N_1P$ and $N_1P \neq N_2P$,
(iii) $P \neq N_1P$ and $N_1P = N_2P$,
(iv) $P \neq N_1P$ and $N_1P \neq N_2P$.

We are not only interested in the question, whether for each of these possibilities a structure with this possibility exists, we are also interested in the signatures of such structures. The following proposition shows, that the difficulty is to minimize the signature, since there are cases, where the converse of this proposition is nontrivial.

PROPOSITION 1. *For each of the above four possibilities: If we have a structure $\mathcal{A}$ of some signature s, which this possibility holds over, then for any signature s' larger than s, there is a structure of signature s', which this possibility holds over.*

PROOF. Extend our structure $\mathcal{A}$ of signature s to a structure of signature s' with the desired property by adding appropriately many copies of the empty set as new relations. ⊣

We do not know a structure of finite signature with (i) but we give an example of countably infinite signature, Theorem 1. For case (ii) we have an example of finite signature, Theorem 2. No natural examples of structures (of any signature) are known for cases (i) and (ii). We do not present a structure with case (iii) in this paper, however in [10] it is proved that $(\mathbf{R}, +, -, 0, 1)$ is a natural example of finite signature. For case (iv), we construct a further structure of finite signature, Theorem 3. Here natural examples of finite signature exist. Moreover, in this section we prove two lemmas also useful for later sections.

The next lemma shows that over certain structures we can code input strings by elements of the structure in linear time. This is interesting because input strings can be of arbitrary finite length whereas the tuples contained in the relation of a structure are of some fixed length.

LEMMA 1. *Let $\mathscr{A}$ be a structure with universe* $\mathbf{N}$, 0 *as a constant, containing the successor function on* $\mathbf{N}$ *and a bijection* π' *from* $\mathbf{N} \times \mathbf{N}$ *to* $\mathbf{N}$ *as one of its functions. Then there exists a bijection* π *from* $\mathbf{N}^+$ *to* $\mathbf{N}$ *which is computable in linear time over* $\mathscr{A}$ *by a program containing only those symbols of the language for* $\mathscr{A}$ *corresponding to* 0, π', *and the successor function.*

PROOF. By induction on k define the bijections $\pi^k\colon \mathbf{N}^k \to \mathbf{N}$ by $\pi^1(n) = n$ and $\pi^{k+1}(n_1, \dots, n_{k+1}) = \pi'(n_1, \pi^k(n_2, \dots, n_{k+1}))$. For a string $z \in \mathbf{N}^+$ of length k define $\pi(z) = \pi'(k-1, \pi^k(z))$. Then π is a bijection $\mathbf{N}^+ \to \mathbf{N}$. Moreover, π is computable in linear time over $\mathscr{A}$ since we can compute $k-1$ in a tape cell for z of length k by moving along z and counting the moves starting from 0 using the successor function. ⊣

Now we turn to the first possible case, namely, that $P = N_2P$.

THEOREM 1. *There is a structure* $\mathscr{A}$ *of countably infinite signature with* $P = N_2P$. *Moreover, we can even arrange, that any set recognizable by some program over* $\mathscr{A}$ *can deterministically be recognized in linear time.*

PROOF. The universe of $\mathscr{A}$ is $\mathbf{N}$. The language for $\mathscr{A}$ contains one constant symbol, interpreted as 0, one unary function symbol, interpreted as the successor function, a binary function symbol, interpreted as a bijection from $\mathbf{N} \times \mathbf{N}$ to $\mathbf{N}$, and an infinite sequence $X_0, X_1, \dots$ of unary relation symbols which we interpret as follows.

Let $N_0^*, N_1^*, \dots$ be an enumeration of all programs over this language and π according to Lemma 1 for $\mathscr{A}$, which we can now already determine. Do for $i = 0, 1, \dots$ in increasing order of i the following: Interpret those X_j appearing in N_i^* and not yet interpreted by the empty set. So, the set

$$L(N_i^*) = \{z \in \mathbf{N}^+ \colon N_i^* \text{ accepts } z \text{ over } \mathscr{A}\}$$

is now determined. Let $X_{k(i)}$ be the relation symbol with least index, which is now not yet interpreted. Interpret it by $\{\pi(w) \colon w \in L(N_i^*)\}$. This concludes the construction of $\mathscr{A}$.

The set recognized by N_l^* over $\mathscr{A}$ can hence deterministically be recognized in linear time over $\mathscr{A}$ by a deterministic program M as follows: On input $z \in \mathbf{N}^+$ compute $\pi(z)$ and check, whether $\pi(z) \in X_{k(l)}^{\mathscr{A}}$. Here $k(l)$ is a fixed number, that does not have to be computed, namely $X_{k(l)}$ is a fixed relation symbol appearing in M. ⊣

It is easy to ensure in the above proof that $X_{k(i)}^{\mathscr{A}}$ becomes classically recursively enumerable in $\emptyset^{(i)}$ uniformly in i, where $\emptyset^{(i)}$ is the ith classical jump of the empty set, $\emptyset^{(0)} := \emptyset$.

The statement of the next lemma is analogous to the fact that one can "clock" Turing machines in classical complexity theory.

LEMMA 2. *Let* $\mathscr{L}$ *be a language of cardinality* $\leq \omega$.

(i) *Then there is a sequence* $M_0, M_1, \dots$ *of deterministic* $\mathscr{L}$*-programs and a sequence* $q_0', q_1', \dots$ *of polynomials, such that over each* $\mathscr{L}$*-structure we have:*
First, the sets L *recognized by the* M_i *are exactly the* $L \in P$.

Second, for each M_i, the computation of M_i on an input of length n has length at most $q_i'(n)$.

(ii) *Second, there is a sequence $N_0, N_1, \dots$ of $\mathcal{L}$-programs nondeterministic of the first kind and a sequence $q_0, q_1, \dots$ of polynomials with $q_i(n) \geq i$ and $lim_n\, q_i(n) = \infty$, such that over each $\mathcal{L}$-structure the following holds:*
First, the sets L recognized by the N_i are exactly the $L \in N_1P$.
Second, for each N_i each computation of N_i on an input of length n has length at most $q_i(n)$.

(iii) *Third, there is an enumeration $\tilde{N}_0, \tilde{N}_1, \dots$ of $\mathcal{L}$-programs nondeterministic of the second kind and a sequence $\tilde{q}_0, \tilde{q}_1, \dots$ of polynomials, such that over each $\mathcal{L}$-structure we have:*
First, $\{L(\tilde{N}_i): i \geq 0\} = N_2P$ and for all i, $N_2TIME(n^i) = \{L(\tilde{N}_j): \tilde{q}_j \in O(n^i)\}$.
Second, for each $\tilde{N}_i$, each computation of $\tilde{N}_i(w)$ has length $\leq \tilde{q}_i(|w|)$.

PROOF. (i) In classical complexity theory one clocks appropriately an enumeration of all deterministic TMs to get an enumeration of machines satisfying the conclusions of (i) for the classical case. Here we can proceed similarly. First we generate a substring y on the tape of length the number of steps to be at most simulated. Note that we can generate a substring of length $k \cdot l$ given substrings of length k and of length l by producing k concatenated versions of the substring of length l. We mark the ends of y by two pointers. We ensure that the distance between these two pointers stays constant during the whole further computation though the substring marked by these pointers can change. Then we move from one pointer to the other with a third pointer, for each cell thereby scanned we simulate one computation step and we stop if we have reached the other pointer. The tape content of the computation to be simulated is a substring of the whole tape content and we mark the ends of this substring by two further pointers. So we can simulate one computation step using only constantly many computation steps. Hence if we simulate $t(n)$ steps, the whole simulation takes $O(t(n))$ steps.

The proof of (ii) and (iii) is similar to the proof of (i). ⊣

Remark: Over any structure $(A, \mathcal{I})$ (of any signature) $L \in P$ iff $A^+ \setminus L \in P$. To show this, let $(A, \mathcal{I})$ be given. Using the proof of Lemma 2(i) we see that for all $L \in P$ there is a deterministic $\mathcal{L}$-program M and a polynomial q', such that M recognizes L and such that the computation of M on an input of length n has length at most $q'(n)$. From M we can obtain a deterministic $\mathcal{L}$-program M' recognizing $A^+ \setminus L$, such that the computation of M' on an input of length n has length at most $q'(n) + 1$. M' can be obtained from M by turning accepting stop instructions into stop instructions which are not accepting and so on.

Now we turn to our example of a structure of finite signature with possibility (ii). The existence of such a structure is particularly interesting because some examples of structures of finite signature with $P \neq N_1P$ are known, see e.g., [12].

In view of the proof of the next theorem we make the following remark. We can construct a set $A \subseteq \mathbf{N}$ in stages s, $s \geq 0$, as follows. At the beginning of the construction it is undetermined for all $j \in \mathbf{N}$, whether $j \in A$ shall hold. Then during stage s we pick some numbers $j \in \mathbf{N}$, such that it is not yet determined, whether $j \in A$ shall hold and determine this for each such j. If we pick each j

during some stage, we have uniquely defined a set $A \subseteq \mathbf{N}$. This approach can also be described in terms of characteristic functions. We want to construct a set $A \subseteq \mathbf{N}$, equivalently its characteristic function $\chi_A : \mathbf{N} \to \{0, 1\}$ in stages s. We start with the nowhere defined function with empty domain, which is a partial function $\mathbf{N} \to \{0, 1\}$. Then at the end of stage s we have constructed a partial function $f_s : \mathbf{N} \to \{0, 1\}$ and extend it to a partial function $f_{s+1} : \mathbf{N} \to \{0, 1\}$ in stage $s + 1$. If we ensure that $\bigcup_s f_s$ is total we can let $\chi_A = \bigcup_s f_s$. We can also construct subsets of e.g., $\mathbf{N}^2$ or $\mathbf{N}^3$ this way and we obtain a clearer proof in doing so concerning $R^{\mathscr{A}}$ and $Q^{\mathscr{A}}$ in the next proof.

THEOREM 2. *There is a structure $\mathscr{A}$ of finite signature with $P = N_1P$ and $N_1P \neq N_2P$.*

PROOF. As a preparation to the proof note that there is a bijection $\pi' : \mathbf{N} \times \mathbf{N} \to \mathbf{N}$, such that $\pi'(n, m) \leq (max\{n, m\} + 1)^2 - 1$. The function π' exists, since there are exactly $(n + 1)^2$ pairs $(k, l) \in \mathbf{N} \times \mathbf{N}$ with $k, l \leq n$. So by induction on n we can construct π', such that $\pi'(k, l) \leq (n + 1)^2 - 1$ if $k, l \leq n$.

Now the universe of $\mathscr{A}$ is $\mathbf{N}$. Let $\mathscr{A}$ contain 0, S, the function f with $f(m) = (m + 1)^2$, the above π', and $\leq$. Besides the corresponding symbols the language $\mathscr{L}$ for $\mathscr{A}$ contains a binary relation symbol Q and a three-placed relation symbol R.

Let $N_0, N_1, \ldots$ be a sequence of $\mathscr{L}$-programs nondeterministic of the first kind and $q_0, q_1, \ldots$ be a sequence of polynomials according to Lemma 2(ii).

Then S, π', 0 determine a bijection $\pi : \mathbf{N}^+ \to \mathbf{N}$ computable in linear time by Lemma 1.

We need the following properties of f.

LEMMA 3. (i) $S(m) \leq f(m), \pi'(n, m), \pi'(m, n) \leq f(m)$ *if* $n \leq m$.

(ii) *We have* $m < f(m)$ *and* $n \leq m \Rightarrow f(n) \leq f(m)$. ⊣

We next give some preparations for the construction of $Q^{\mathscr{A}}$ and $R^{\mathscr{A}}$. For each pair (i, w), $i \in \mathbf{N}$, $w \in \mathbf{N}^+$ we define a number $\zeta(i, w)$ as follows. Let $w = (a_1, \ldots, a_n)$ and $m = max\{a_1, \ldots, a_n\}$. Then $\zeta(i, w) = f^{q_i(n)}(m)$, the $q_i(n)$-fold application of f to m.

LEMMA 4. *For all $s \in \mathbf{N}$ there are at most finitely many pairs (i, w) with $\zeta(i, w) = s$.*

PROOF. Fix $s \in \mathbf{N}$. First we see that there at most finitely many $i \in \mathbf{N}$, such that there is some $w \in \mathbf{N}^+$ with $\zeta(i, w) = s$: For all $i > s$ we have for all $w \in \mathbf{N}^+$ that $\zeta(i, w) > s$ by definition of $\zeta(i, w)$, f and by $q_i(n) \geq i$.

So it suffices to show, that for each i there are at most finitely many w with $\zeta(i, w) = s$. So fix i. We first see that there are at most finitely many n, such that there is a $w \in \mathbf{N}^+$ of length n with $\zeta(i, w) = s$. Since $lim_n\, q_i(n) = \infty$ choose n_0 such that for all $n \geq n_0$, $q_i(n) > s$. Then for all w of length $n \geq n_0$ we have $\zeta(i, w) > s$. Hence, to complete the proof we fix i and n and see that there are at most finitely many w of length n with $\zeta(i, w) = s$. There are only finitely many $(a_1, \ldots, a_n)$ with $m = max\{a_1, \ldots, a_n\} \leq s$ but if $m > s$ then $\zeta(i, w) > s$. ⊣

By Lemma 4 there is a sequence $(z_h)_{h \geq 0}$ of pairs (i, w), $i \in \mathbf{N}$, $w \in \mathbf{N}^+$, in which each (i, w) appears exactly once, such that

$$\zeta(z_0) \leq \zeta(z_1) \leq \ldots$$

Now the following simultaneous construction of $Q^{\mathscr{A}}$ and $R^{\mathscr{A}}$ is done by fixing for more and more pairs (n, m), whether $(n, m) \in Q^{\mathscr{A}}$ shall hold and for more and more triples (n, m, l) whether $(n, m, l) \in R^{\mathscr{A}}$ shall hold, initially this is undetermined for all pairs and triples. During this construction we simulate programs N_i on certain inputs over our structure $\mathscr{A}$ to be constructed, more precisely, we generate sequences of configurations which will be computations of the programs on the corresponding inputs over $\mathscr{A}$. If we simulate $N_i(y)$, then we generate all possible computations of $N_i(y)$. Here it can happen, that we have to know, whether $(n, m) \in Q^{\mathscr{A}}$ holds or that we have to know, whether $(n, m, l) \in R^{\mathscr{A}}$ holds for some n, m, l though we have not yet fixed this. Then we define $(n, m) \notin Q^{\mathscr{A}}$ or $(n, m, l) \notin R^{\mathscr{A}}$, respectively.

During the construction we keep a list I of indices $i \in \mathbf{N}$ and a list J of strings $w \in \mathbf{N}^+$. Initially $I = J = \emptyset$. Perform in increasing order of s the following stages.

Construction.

Stage s. Let $i \in \mathbf{N}$ be the number and $w \in \mathbf{N}^+$ be the string with $(i, w) = z_s$. Let $w = (a_1, \dots, a_n)$ and $m = max\{a_1, \dots, a_n\}$.

First part. Simulate $N_i(w)$ as described above.

Second part. (For $P = N_1 P$). Let $t = f^{q_i(n)+1}(m)$, the $(q_i(n) + 1)$-fold application of f to m. If we have not yet determined, whether $(i, \pi(w), t) \in R^{\mathscr{A}}$, let $(i, \pi(w), t) \in R^{\mathscr{A}}$ if additionally $N_i(w)$ accepts and $(i, \pi(w), t) \notin R^{\mathscr{A}}$ if additionally $N_i(w)$ does not accept, (else do nothing).

Third part. (For $N_1 P \neq N_2 P$). If $i = max\, I + 1$, $I \neq \emptyset$, and $w \notin J$ or if $i = 0$ and $I = \emptyset$ then put i into I and w into J and do the following. Choose the least k, such that we have not yet determined, whether $(\pi(w), k) \in Q^{\mathscr{A}}$ shall hold. If $N_i(w)$ does not accept, let $(\pi(w), k) \in Q^{\mathscr{A}}$. In all other cases do nothing. This finishes the construction.

Clearly, the construction can be performed successfully, since at any point during the construction we have determined, whether $(h, k) \in Q^{\mathscr{A}}$ shall hold for only finitely many pairs (h, k). Further, for all h, l, k we have determined, whether $(h, l, k) \in R^{\mathscr{A}}$ holds during the construction. Namely, we have determined this at the end of stage s, if $z_s = (i, w)$, such that i and w have the following properties. $w = (h, l, k)$ and N_i is a program which checks on input (h, l, k), whether $(h, l, k) \in R^{\mathscr{A}}$ holds. Similarly, for all h, l we have determined, whether $(h, l) \in Q^{\mathscr{A}}$ holds. The next two lemmas prepare the proof that $P = N_1 P$ over $\mathscr{A}$.

Lemma 5. *For all $h, i \in \mathbf{N}$ and $w \in \mathbf{N}^+$: If h appears in a tape cell in some computation of $N_i(w)$, then $h \leq \zeta(i, w)$.*

Proof. Let $h, i \in \mathbf{N}$, $w \in \mathbf{N}^+$, $w = (a_1, \dots, a_n)$, and a computation p of $N_i(w)$ of some length $k \leq q_i(n)$ be given. Recall that 0 is the only constant. Using Lemma 3, by induction we see that for $1 \leq l \leq k$ all numbers appearing in a tape cell in the lth configuration of p are $\leq f^{l-1}(m)$, $m = max\{a_1, \dots, a_n\}$. So again by Lemma 3, Lemma 5 follows. ⊣

Lemma 6. *For all s, at the beginning of the second part in stage s we have not yet determined, whether $(i, \pi(w), t) \in R^{\mathscr{A}}$ holds.*

Proof. Fix s with corresponding i, w, and t. Let $m = max\{a_1, \dots, a_n\}$, where $w = (a_1, \dots, a_n)$. First we see that we have not yet determined, whether $(i, \pi(w), t) \in R^{\mathscr{A}}$ holds in the first part of a stage $s' \leq s$. By definition of the sequence $(z_h)_{h \geq 0}$ and Lemma 5, if we determine, whether $(d, k, l) \in R^{\mathscr{A}}$ holds in the first part of a stage $s' \leq s$, then $l \leq \zeta(i, w)$. But by Lemma 3

$$t = f^{q_i(n)+1}(m) > f^{q_i(n)}(m) = \zeta(i, w).$$

Second, we have not yet determined, whether $(i, \pi(w), t) \in R^{\mathscr{A}}$ holds in the second part of a stage $s' < s$. Since if $s' < s$, for $(i', w') = z_{s'}$ we have $(i', w') \neq (i, w)$, so $(i, \pi(w)) \neq (i', \pi(w'))$. Clearly, in the third part of a stage $s' < s$ we have not yet determined this, too. ⊣

Using Lemma 6, we see that for all $i \in \mathbf{N}$ and $w \in \mathbf{N}^+$, $w = (a_1, \dots, a_n)$, $N_i(w)$ accepts iff $(i, \pi(w), t) \in R^{\mathscr{A}}$, where $t = f^{q_i(n)+1}(m)$, $m = max\{a_1, \dots, a_n\}$. Hence $L(N_i)$ can deterministically be recognized in polynomial time as follows. Let $w = (a_1, \dots, a_n)$ be the input. Compute i using 0 and S and compute $\pi(w)$ and $m = max\{a_1, \dots, a_n\}$ in a tape cell. Similarly as in the proof of Lemma 2, generate a substring of length $q_i(n) + 1$ on the tape and compute $t = f^{q_i(n)+1}(m)$ by iterating f $(q_i(n) + 1)$-times. Finally, check whether $(i, \pi(w), t) \in R^{\mathscr{A}}$. So $P = N_1P$.

It remains to see $N_1P \neq N_2P$. Define L by

$$w \in L \Leftrightarrow \exists k\ (\pi(w), k) \in Q^{\mathscr{A}}.$$

Then $L \in N_2P$. To see $L \notin N_1P$, we first see that it follows by induction on i that for all i there is an s, such that at the end of stage s, $I = \{0, \dots, i\}$ and $J = \{w(0), \dots, w(i)\}$ for some $w(0), \dots, w(i) \in \mathbf{N}^+$.

$\underline{i = 0}$: Choose s minimal, such that $z_s = (0, w)$ for some $w \in \mathbf{N}^+$. Let $w(0) = w$. Then $I = J = \emptyset$ at the beginning of the third part in stage s. So $I = \{0\}$ and $J = \{w(0)\}$ at the end of stage s.

$\underline{i \to i + 1}$: Let s be given, such that at the end of stage s we have $I = \{0, \dots, i\}$ and $J = \{w(0), \dots, w(i)\}$ for some $w(0), \dots, w(i) \in \mathbf{N}^+$. Choose $s' > s$ minimal such that $z_{s'} = (i+1, w)$ for some $w \in \mathbf{N}^+$ with $w \notin \{w(0), \dots, w(i)\}$. Let $w(i+1) = w$. Then at the end of stage s' $I = \{0, \dots, i + 1\}$, $J = \{w(0), \dots, w(i + 1)\}$.

Now we can prove $L \neq L(N_i)$ for all i. So fix i. By the above, i is put into I in some stage s and the corresponding y with $z_s = (i, y)$ is put into J. So by construction, $y \notin J$ at the beginning of the third part in stage s, whence at this point there is no k, such that we have already determined $(\pi(y), k) \in Q^{\mathscr{A}}$. Then during that third part we let $(\pi(y), k) \in Q^{\mathscr{A}}$ for some k iff $N_i(y)$ does not accept. After stage s, we let $(\pi(y), k) \in Q^{\mathscr{A}}$ for no k since then $y \in J$. Hence

$$y \in L \Leftrightarrow \exists k\ (\pi(y), k) \in Q^{\mathscr{A}} \Leftrightarrow N_i(y) \text{ does not accept.}$$

So $L \neq L(N_i)$ and $N_1P \neq N_2P$. This proves Theorem 2. ⊣

In the above proof we can easily ensure that $Q^{\mathscr{A}}$ and $R^{\mathscr{A}}$ become classically recursive.

Skipping (iii) as stated above, we finally treat case (iv), where $P \neq N_1P$ and $N_1P \neq N_2P$. The unordered group of integers is a natural example of finite signature: The set of even integers viewed as a set of inputs of length 1 belongs to $N_2P \setminus N_1P$ and $P \neq N_1P$ holds over every infinite Abelian group, for a proof of the latter see e.g., [12]. However, in this paper we are interested in the power of computability theoretic constructions, so we prefer a proof of the existence of a structure with $P \neq N_1P$ and $N_1P \neq N_2P$ of the following kind. The proof of the next theorem giving such a structure of finite signature uses only two simple diagonalizations, one for $P \neq N_1P$ and one for $N_1P \neq N_2P$.

THEOREM 3. *There is a structure $\mathscr{A}$ of finite signature with $P \neq N_1P$ and $N_1P \neq N_2P$.*

PROOF. The universe of $\mathscr{A}$ is $\mathbf{N}$. Let $\mathscr{A}$ contain the successor function, the addition, 2^n, and a bijection from $\mathbf{N} \times \mathbf{N}$ to $\mathbf{N}$ as its functions. 0 is its only constant. The language $\mathscr{L}$ of $\mathscr{A}$ contains as relation symbols two binary symbols Q and R. This determines a function π for $\mathscr{A}$ according to Lemma 1.

As in Lemma 2 we have an enumeration $N_0, N_1, \dots$ of programs and an enumeration M_0, $M_1, \dots$ of programs with corresponding polynomials q_i'.

We will arrange that $L \in N_1P \setminus P$, where L is defined as follows:

$$w \in L \Leftrightarrow \exists j < 2^{|w|}\ (\pi(w), j) \in Q^{\mathscr{A}}.$$

It will hold $L' \in N_2P \setminus N_1P$, L' consisting of strings of length 1 with

$$(n) \in L' \Leftrightarrow \exists j\ (n, j) \in R^{\mathscr{A}}.$$

Now we turn to the construction of $Q^{\mathscr{A}}$ and $R^{\mathscr{A}}$. Similarly as in the proof of Theorem 2 we do this by fixing for more and more pairs (n, m), whether $(n, m) \in Q^{\mathscr{A}}$ or $(n, m) \in R^{\mathscr{A}}$ shall hold, respectively. Initially it is undefined for all (n, m), whether $(n, m) \in Q^{\mathscr{A}}$ shall hold and whether $(n, m) \in R^{\mathscr{A}}$ shall hold. Again similarly as in the proof of Theorem 2, during this construction we simulate programs N_i or M_i on some input over our structure $\mathscr{A}$ to be constructed, more precisely, we generate sequences of configurations which will be computations of the programs on the corresponding inputs over $\mathscr{A}$. In case of M_i on some input we generate a unique sequence and in case of N_i on some input we generate all possible computations over $\mathscr{A}$. Here it can happen, that we have to know, whether $(n, m) \in Q^{\mathscr{A}}$ holds or that we have to know whether $(n, m) \in R^{\mathscr{A}}$ holds for some (n, m) though we have not yet fixed this. Then we define $(n, m) \notin Q^{\mathscr{A}}$ or $(n, m) \notin R^{\mathscr{A}}$, respectively.

We remark, that at each point in the construction, there are only finitely many pairs (n, m), such that we have fixed, whether $(n, m) \in Q^{\mathscr{A}}$ shall hold, the same for $R^{\mathscr{A}}$. Perform in increasing order of i the following stages.

Construction.

Stage i. (For $P \neq N_1P$). Choose n_i, $n_i \neq n_{i'}$ for $i' < i$ such that $2^{n_i} > q_i'(n_i)$ and such that it is not yet fixed for all j, whether $(\pi(0^{n_i}), j) \in Q^{\mathscr{A}}$ holds. Simulate $M_i(0^{n_i})$ as described above. If the simulated computation is not accepting, let $(\pi(0^{n_i}), j) \in Q^{\mathscr{A}}$ for some $j < 2^{n_i}$, which it is not yet fixed for, whether $(\pi(0^{n_i}), j) \in Q^{\mathscr{A}}$ holds, (else do nothing).

(For $N_1P \neq N_2P$). Simulate $N_i(i)$ as described above. If no accepting computation has been simulated, let $(i, j) \in R^{\mathscr{A}}$ for some j, which it is not yet fixed for, whether $(i, j) \in R^{\mathscr{A}}$ holds, (else do nothing). This finishes the construction.

Using the remark before the construction, we see that the construction can be performed successfully. Moreover, $Q^{\mathscr{A}}$ and $R^{\mathscr{A}}$ are defined completely during the construction: Since we are given the constant 0 and the successor function in $\mathscr{A}$ ahead of the construction, we can generate each $j \in \mathbf{N}$ in a tape cell on arbitrary input, so for each pair (n, m) there is some M_i, such that we need, whether $(n, m) \in Q^{\mathscr{A}}$ holds for its simulation in the construction; similarly for $R^{\mathscr{A}}$. So we have obtained a well-defined structure $\mathscr{A}$.

Now we show that $N_1P \neq N_2P$ holds over $\mathscr{A}$. Clearly, $L' \in N_2P$. To see $L' \notin N_1P$, note that by construction for all i, $N_i(i)$ does not accept if only if $(i, j) \in R^{\mathscr{A}}$ for some j if and only if $(i) \in L'$.

Finally, we see that $P \neq N_1P$ over $\mathscr{A}$. First, we check that $L \in N_1P$. Note that for each n, each $0 \leq m < 2^n$ is of the form $\sum_{i=0}^{n-1} a_i 2^i$ with $a_i \in \{0, 1\}$. Hence, on input w, we first compute nondeterministically a number $0 \leq m < 2^{|w|}$, such that each such number is computed in some computation, where the nondeterministic steps correspond to choosing $a_i = 0$ or $a_i = 1$. Then we can deterministically compute $\pi(w)$ and check, whether $(\pi(w), m) \in Q^{\mathscr{A}}$ holds. Further, $L \notin P$, since by construction for all i, $M_i(0^{n_i})$ does not accept if and only if $(\pi(0^{n_i}), j) \in Q^{\mathscr{A}}$ for some $j < 2^{n_i}$ if and only if $0^{n_i} \in L$. ⊣

In the above proof we can easily ensure that $Q^{\mathscr{A}}$ and $R^{\mathscr{A}}$ become classically recursive.

§3. **$P = N_1P$ and ultraproducts.** Structures are also considered from the point of view of model theory and so we address the question, which connections there are between model theory and computability theory. In this section we want to give an example of such a connection, which is based on the proof of Theorem 2. For ultraproducts, substructures, and other notions of model theory used in this section see [3] or [4].

Ultraproducts are a fundamental construction in model theory, different from the computability theoretic constructions of structures introduced in the previous section. In this section we give a proof in which both kinds of constructions are combined.

Questions of whether certain classes of structures are closed under certain operations are important in model theory, and the notions of Δ-elementary classes and of elementary classes are of interest, where a class $\mathscr{K}$ of $\mathscr{L}$-structures is a Δ-elementary class iff there exists a theory T in $\mathscr{L}$ such that $\mathscr{K}$ is exactly the class of all models of T. If we can choose $T = \{\varphi\}$ for some $\mathscr{L}$-sentence φ, then $\mathscr{K}$ is elementary. The classes of torsion-free Abelian groups, fields of characteristic zero etc. can all be obtained as Δ-elementary classes for some appropriate $\mathscr{L}$, and the classes of groups, Abelian groups, fields etc. as elementary classes. One can show that a theory T has a set of axioms with certain syntactical properties iff it is preserved under certain operations on structures like unions of chains or homomorphisms, or for example one can prove that a sentence φ is preserved under reduced products iff it is equivalent to a Horn sentence.

It is further known, that a class $\mathscr{K}$ of $\mathscr{L}$-structures is a Δ-elementary class iff $\mathscr{K}$ is closed under ultraproducts and elementary equivalence, and that $\mathscr{K}$ is elementary iff $\mathscr{K}$ and the class of $\mathscr{L}$-structures not in $\mathscr{K}$ are Δ-elementary. We remark that for each $\mathscr{L}$-structure $\mathscr{A}$ with $P = N_1P$ there is a set T of universal $\mathscr{L}$-sentences, such that $\mathscr{A}$ is a model of T and such that over each other $\mathscr{L}$-structure which is a model of T we also have $P = N_1P$. So for all $\mathscr{L}$, the class of $\mathscr{L}$-structures with $P = N_1P$ is the union of some Δ-elementary classes, equivalently is closed under elementary equivalence. Hence it resembles a Δ-elementary class and we ask, whether it is Δ-elementary and closed under ultraproducts. We prove in this section that for some finite $\mathscr{L}$ the class of $\mathscr{L}$-structures with $P = N_1P$ is not closed under ultraproducts. As a corollary we obtain, that this class is not a Δ-elementary class, though some Δ-elementary classes arise in computability theory over structures as we will see. This corollary implies that the class of $\mathscr{L}$-structures with $P \neq N_1 P$ is not elementary. These facts can also be proved using the compactness theorem.

We further remark that the class of $\mathscr{L}$-structures with $P = N_1 P$ is closed under substructures for all $\mathscr{L}$. However, using the ideas to construct structures with $P = N_1 P$ and $P \neq N_1 P$ from the previous section it follows that for some $\mathscr{L}$ this class is not closed under extensions and homomorphisms, the latter even if the structures have the same universe and the homomorphism is the identity on the common universe.

THEOREM 4. *There is a sequence $\mathscr{A}_j$, $j \in \mathbf{N}$, of $\mathscr{L}$-structures for some finite $\mathscr{L}$ with $P = N_1 P$ and an ultrafilter D over $\mathbf{N}$, such that $P \neq N_1 P$ over $\Pi_D \mathscr{A}_j$. Hence the class of structures of finite signature with $P = N_1 P$ is not closed under ultraproducts.*

PROOF. Let $\mathscr{L} = \{F, K, H, O, U, Q, R, c\}$, where F, K are unary function symbols, H, O are binary function symbols, U, Q are binary relation symbols, R is a three-placed relation symbol and c is a constant symbol.

Let $N_0, N_1, \dots$ be a sequence of $\mathscr{L}$-programs nondeterministic of the first kind and $q_0, q_1, \dots$ be a sequence of polynomials according to Lemma 2(ii) and let $M_0, M_1, \dots$ be a sequence of deterministic $\mathscr{L}$-programs and $q'_0, q'_1, \dots$ be a sequence of polynomials according to Lemma 2(i).

As in the proof of Theorem 2 let $\pi' \colon \mathbf{N} \times \mathbf{N} \to \mathbf{N}$ be a bijection, such that $\pi'(n, m) \leq (max\{n, m\} + 1)^2 - 1$. Let $f \colon \mathbf{N} \to \mathbf{N}$ be the function with $f(m) = 2^{m+1}$.

Next we define the structures $\mathscr{A}_j$. So fix $j \in \mathbf{N}$. The universe of $\mathscr{A}_j$ is $\mathbf{N}$. Let $F^{\mathscr{A}_j} = S$, $K^{\mathscr{A}_j} = f$, $H^{\mathscr{A}_j} = +$, $O^{\mathscr{A}_j} = \pi'$, $U^{\mathscr{A}_j}$ is $\leq$, $c^{\mathscr{A}_j} = 0$. $Q^{\mathscr{A}_j}$ and $R^{\mathscr{A}_j}$ are defined later.

Then π', 0, S determine a bijection $\pi \colon \mathbf{N}^+ \to \mathbf{N}$ computable in linear time by Lemma 1. Observe that

LEMMA 7. (i) $S(m), m + m, (m + 1)^2 - 1 \leq f(m)$.

(ii) *We have $m < f(m)$ and $n < m \Rightarrow f(n) < f(m)$.* $\dashv$

For each pair (i, w), $i \in \mathbf{N}$, $w \in \mathbf{N}^+$, we define a number $\zeta(i, w)$ as follows: Let $w = (a_1, \dots, a_n)$ and $m = max\{a_1, \dots, a_n\}$. Then $\zeta(i, w) = f^{q_i(n)}(m)$, the $q_i(n)$-fold application of f to m. With the same proof as for Lemma 4 we have again that for all $s \in \mathbf{N}$ there are at most finitely many pairs (i, w) with $\zeta(i, w) = s$. So again there is a sequence $(z_h)_{h \geq 0}$ of pairs (i, w), $i \in \mathbf{N}$, $w \in \mathbf{N}^+$, in which each (i, w) appears exactly once, such that

$$\zeta(z_0) \leq \zeta(z_1) \leq \dots$$

Now we turn to the construction of $Q^{\mathscr{A}_j}$ and $R^{\mathscr{A}_j}$. We do this by fixing for more and more tuples (n, m) and (n, m, l) whether $(n, m) \in Q^{\mathscr{A}_j}$ and $(n, m, l) \in R^{\mathscr{A}_j}$ shall hold, respectively. Initially this is undefined for all (n, m) and (n, m, l). If we simulate a program N_i or M_i during this construction, then we do this as in the proof of Theorem 2, where in case of M_i we generate one unique computation. The construction consists of two parts, the first part is for $P \neq N_1 P$ over $\Pi_D \mathscr{A}_i$ for some ultrafilter D over $\mathbf{N}$ and the second part ensures that $P = N_1 P$ over $\mathscr{A}_j$.

Construction.

FIRST PART. Do in increasing order of s, $0 \leq s \leq j$, the following. Let n_s be the least $n \geq 1$ such that:

(i) $n \neq n_{s'}$ for $s' < s$,
(ii) for all m we have not yet determined, whether $(\pi(0^n), m) \in Q^{\mathscr{A}_j}$ shall hold,
(iii) $2^n > q'_s(n)$.

Simulate $M_s(0^{n_s})$. If $M_s(0^{n_s})$ does not accept, find m, $0 \leq m < 2^{n_s}$, such that it is not yet fixed, whether $(\pi(0^{n_s}), m) \in Q^{\mathscr{A}_j}$ holds, and let $(\pi(0^{n_s}), m) \in Q^{\mathscr{A}_j}$, (else do nothing).

SECOND PART. Let $k_j > 0$ be a number such that for all h, all l, and all $k \geq k_j$ we have not yet defined, whether $(h, l, k) \in R^{\mathscr{A}_j}$ shall hold. Perform in increasing order of s the following stages, $s \geq 0$.

Stage s. Let $i \in \mathbf{N}$ be the number and $w \in \mathbf{N}^+$ the string with $(i, w) = z_s$.

SUBSTAGE 1. Simulate $N_i(w)$.

SUBSTAGE 2. Let $t = \zeta(i, w) + k_j$. If we have not yet determined, whether $(i, \pi(w), t) \in R^{\mathscr{A}_j}$, let $(i, \pi(w), t) \in R^{\mathscr{A}_j}$ if additionally $N_i(w)$ accepts and $(i, \pi(w), t) \notin R^{\mathscr{A}_j}$ if additionally $N_i(w)$ does not accept, (else do nothing). This finishes the construction.

As in the proof of Theorem 2 we have obtained a well-defined structure. For all j, $P = N_1P$ over $\mathscr{A}_j$ follows along lines of $P = N_1P$ in the proof of Theorem 2, except that for Lemma 6 we have to keep the definition of k_j in mind.

It remains to see that $P \neq N_1P$ over $\Pi_D\mathscr{A}_j$ for some ultrafilter D over $\mathbf{N}$. Let $E = \{\{i : i \geq j\} : j \in \mathbf{N}\}$. Since E has the finite intersection property, there is an ultrafilter D over $\mathbf{N}$, such that $E \subseteq D$, namely D is an ultrafilter containing the Fréchet filter. Fix such a D. We will see that $P \neq N_1P$ over $\Pi_D\mathscr{A}_j$. Let $\mathscr{A} = \Pi_D\mathscr{A}_j$.

First note that for all $i \in \mathbf{N}$ and $n \geq 1$ there is a quantifier-free formula $\psi_{i,n} = \psi_{i,n}(x_1, \dots, x_n)$ of $\mathscr{L}$, such that for all $\mathscr{L}$-structures $\mathscr{B}$ and all elements $b_1, \dots, b_n$ of the universe of $\mathscr{B}$, $M_i(b_1, \dots, b_n)$ accepts iff $\mathscr{B} \models \psi_{i,n}[b_1, \dots, b_n]$. We obtain $\psi_{i,n}$ as follows.

Let $k = max\{u : p_u \text{ or } \tilde{p}_u \text{ appears in } M_i\}$. Let p be a sequence of instructions of M_i of length at most $q'_i(n)$, where each time a conditional instruction appears in p we assign either 0 or 1 to $Cond$. We associate a formula ψ^* with p. If there is no $\mathscr{L}$-structure $(B, \mathscr{I})$ and no $(b_1, \dots, b_n) \in B^+$, such that $M_i(b_1, \dots, b_n)$ executes exactly p over $(B, \mathscr{I})$ then $\psi^* = (x_1 \wedge \neg x_1)$. Otherwise, to get ψ^* we observe by induction on h, $1 \leq h \leq |p|$, that for all h, $1 \leq h \leq |p|$, we have:

(i) There are unique numbers $l, m, n_0, \dots, n_k$ and there are $\mathscr{L}$-terms $r_0(x_1, \dots, x_n)$, $\dots, r_l(x_1, \dots, x_n)$, such that the hth configuration of the computation of $M_i(b_1, \dots, b_n)$ over $\mathscr{B} = (B, \mathscr{I})$ is

$$\left(\left(r_0^{\mathscr{B}}[b_1, \dots, b_n], \dots, r_l^{\mathscr{B}}[b_1, \dots, b_n]\right); m; n_0, \dots, n_k\right),$$

if $\mathscr{B}$ is an $\mathscr{L}$-structure, $b_1, \dots, b_n \in B$, the computation of $M_i(b_1, \dots, b_n)$ over $\mathscr{B}$ has length $\geq h$, and the first $h - 1$ computation steps of $M_i(b_1, \dots, b_n)$ over $\mathscr{B}$ are carried out according to the initial segment of p of length $h - 1$.

(ii) There is an $h'(h)$, $0 \leq h'(h) \leq h - 1$, and a finite sequence $v_h = (\theta_1, \theta_2, \dots, \theta_{h'(h)})$ such that the following holds. v_{h-1} is an initial segment of v_h, where v_0 is the empty sequence. Each θ_q is an atomic $\mathscr{L}$-formula or the negation of an atomic $\mathscr{L}$-formula. Furthermore, if $h'(h) \geq 1$, then $(B, \mathscr{I}) \models (\theta_1 \wedge \dots \wedge \theta_{h'(h)})[b_1, \dots, b_n]$ if and only if $\mathscr{B} = (B, \mathscr{I})$ is an $\mathscr{L}$-structure, $b_1, \dots, b_n \in B$, the computation of $M_i(b_1, \dots, b_n)$ over $\mathscr{B}$ has length $\geq h$, and the first $h - 1$ computation steps of

$M_i(b_1, \dots, b_n)$ over $\mathscr{B}$ are carried out according to the initial segment of p of length $h-1$. If $h'(h) = 0$, i.e., iff v_h is empty, then over all $\mathscr{L}$-structures $\mathscr{B} = (B, \mathscr{I})$, for all $b_1, \dots, b_n \in B$, the computation of $M_i(b_1, \dots, b_n)$ over $\mathscr{B}$ has length $\geq h$, and the first $h-1$ computation steps of $M_i(b_1, \dots, b_n)$ over $\mathscr{B}$ are carried out according to the initial segment of p of length $h-1$.

The most interesting case in the proof of (i) and (ii) is the induction step from h to $h+1$ for (ii), if the hth component of p is a conditional instruction with *Cond* a basic data expression. Exactly in this case we choose $h'(h+1) > h'(h)$, namely $h'(h+1) = h'(h) + 1$ where $h'(1) = 0$. If thereby in p we have assigned 1 to *Cond* we obtain a new atomic formula $\theta_{h'(h+1)}$. If thereby in p we have assigned 0 to *Cond* we obtain a new $\theta_{h'(h+1)}$ which is the negation of an atomic formula. For example, if *Cond* is $\tilde{p}_1 \equiv \tilde{p}_2$ and if we have assigned 0 to *Cond* then $\theta_{h'(h+1)}$ is $\neg r_{n_1} \equiv r_{n_2}$, where r_{n_1}, r_{n_2} are chosen according to (i) for h. To define ψ^*, if $h'(|p|) \geq 1$ we let $\psi^* = (\theta_1 \wedge \dots \wedge \theta_{h'(|p|)})$. If $h'(|p|) = 0$ we let $\psi^* = (x_1 \vee \neg x_1)$. So for all $\mathscr{L}$-structures $(B, \mathscr{I})$, for all $b_1, \dots, b_n \in B$, $(B, \mathscr{I}) \models \psi^*[b_1, \dots, b_n]$ iff $M_i(b_1, \dots, b_n)$ executes exactly p over $(B, \mathscr{I})$.

Now we can define $\psi_{i,n}$. The formula $\psi_{i,n}$ is the disjunction of all ψ^* with p ending in an accepting stop instruction; if the disjunction is empty then $\psi_{i,n}$ is $(x_1 \wedge \neg x_1)$.

Similarly, for all $i \in \mathbf{N}$ and $n \geq 1$ we can construct a quantifier-free formula $\varphi_{i,n}(x_1, \dots, x_n)$ of $\mathscr{L}$, such that for all $\mathscr{L}$-structures $\mathscr{B}$ and all elements $b_1, \dots, b_n$ of the universe of $\mathscr{B}$, $N_i(b_1, \dots, b_n)$ accepts iff $\mathscr{B} \models \varphi_{i,n}[b_1, \dots, b_n]$. For such formulas see also [9].

Further, let L_j be the set of all $w \in \mathbf{N}^+$, $w = (a_1, \dots, a_n)$, such that for some m, $0 \leq m < 2^n$, $(\pi(w), m) \in Q^{\mathscr{A}_j}$. By construction of the sequence $(N_i)_{i \geq 0}$, there is an N_l such that over all $\mathscr{A}_j$ we have $L(N_l) = L_j$. N_l can guess a number m, $0 \leq m < 2^n$, using f, $+$, S, 0; recall the binary expansion of natural numbers.

We show that over $\mathscr{A}$ we have $L(N_l) \neq L(M_i)$ for all i, hence $P \neq N_1P$ over $\mathscr{A}$. So fix i. By the above construction there is an $n \geq 1$, such that for all $j \geq i$

$$M_i(0^n) \text{ accepts over } \mathscr{A}_j \Leftrightarrow 0^n \notin L_j.$$

Hence $M_i(0^n)$ accepts over $\mathscr{A}_j$ iff $N_l(0^n)$ does not accept over $\mathscr{A}_j$, $j \geq i$. We have $\mathscr{A}_j \models (\psi_{i,n} \leftrightarrow \neg\varphi_{l,n})[0, \dots, 0]$, $j \geq i$. Hence we further have $\mathscr{A}_j \models \gamma$ for $j \geq i$, where $\gamma = (\psi_{i,n} \leftrightarrow \neg\varphi_{l,n})(c, \dots, c)$ is a sentence of $\mathscr{L}$. So $\{j \in \mathbf{N} \colon \mathscr{A}_j \models \gamma\} \in D$ and $\mathscr{A} \models \gamma$. This means that over $\mathscr{A}$ the program M_i accepts the input $(c^{\mathscr{A}}, \dots, c^{\mathscr{A}})$ of length n iff N_l does not accept this input. So $L(M_i) \neq L(N_l)$ over $\mathscr{A}$. ⊣

It is easy to modify the above proof such that $R^{\mathscr{A}_j}$ and $Q^{\mathscr{A}_j}$ become classically uniformly recursive in j. We can also achieve $N_1P \neq N_2P$ over all $\mathscr{A}_j$ and over $\mathscr{A}$ if we appropriately use the method from the proof of Theorem 2 for $N_1P \neq N_2P$ in the construction of the $\mathscr{A}_j$.

Recalling how we have obtained $\varphi_{i,n}$ and $\psi_{i,n}$ in the above proof, we see that for an arbitrary language $\mathscr{L}$ and an arbitrary $\mathscr{L}$-program N there is an existential $\mathscr{L}$-formula $\delta_{n,k} = \delta_{n,k}(x_1, \dots, x_n)$, such that for all $\mathscr{L}$-structures $(A, \mathscr{I})$ and $a_1, \dots, a_n \in A$ $(A, \mathscr{I}) \models \delta_{n,k}[a_1, \dots, a_n]$ iff $N(a_1, \dots, a_n)$ has a computation of length $> k$. So $(A, \mathscr{I}) \models \forall x_1 \dots \forall x_n \neg\delta_{n,k}$ iff $N(w)$ does not have computations of length $> k$ for $|w| = n$. It follows that for all $f : \mathbf{N} \to \mathbf{N}$ the class of $\mathscr{L}$-structures, such that the computations of $N(w)$ are of length $\leq f(|w|)$ is Δ-elementary. Similarly, the classes of $\mathscr{L}$-structures, such that $L(N) = \emptyset$ or such that N has only infinite computations on arbitrary inputs are Δ-elementary. Such examples are already presented in [9]. In contrast, we have

COROLLARY 1. (i) *For the $\mathscr{L}$ from the above proof, the class of $\mathscr{L}$-structures with $P = N_1P$ is not a Δ-elementary class.*

(ii) *For the $\mathscr{L}$ from the above proof, the class of $\mathscr{L}$-structures with $P \neq N_1P$ is not an elementary class.*

PROOF. (i) By the fundamental theorem on ultraproducts, Δ-elementary classes are closed under ultraproducts.

(ii) If the $\mathscr{L}$-structures with $P \neq N_1P$ were exactly the models of some $\mathscr{L}$-sentence φ, then the $\mathscr{L}$-structures with $P = N_1P$ would be exactly the models of $\neg\varphi$, contradicting (i). $\dashv$

Theorem 4 and its corollary also hold for all $\mathscr{L}'$ with $\mathscr{L} \subseteq \mathscr{L}'$. In the proof of Theorem 4, interpret the symbols from $\mathscr{L}$ as in $\mathscr{A}_j$, interpret new l-ary function symbols by the function g with $g(a_1, \dots, a_l) = a_1$, interpret new relation symbols by $\emptyset$ and interpret new constant symbols by 0. Of course, this does not prove the theorem and its corollary for arbitrary languages, but note that the number of symbols in $\mathscr{L}$ can be reduced and their arities can be varied.

§4. A structure with special properties of N_2P.

The internal structure of NP and the relationship of NP to other complexity classes are an important topic of classical complexity theory. For example, it is known that $P \subseteq NP \subseteq EXP = DTIME(2^{poly})$ and that at least one of these inclusions is proper, but open, whether $P \neq NP$ or $NP \neq EXP$ holds, respectively. Further note that, for certain functions like $f(n) = n^{\lfloor log(n) \rfloor}$, if $DTIME(f) \subseteq NP$, then $P \neq NP$. In the next theorem we consider questions of this kind over structures.

THEOREM 5. *Let $f\colon \mathbf{N} \to \mathbf{N}$ dominate all polynomials, e.g., $f(n) = n^{\lfloor log(n) \rfloor}$. Then there is a structure $\mathscr{A}$ of finite signature with the following properties:*

(i) $P \neq N_1P$,

(ii) $DTIME(n^i) \nsubseteq N_2TIME(n^{i'})$ *if* $i > i'$,

(iii) $DTIME(f) \nsubseteq N_2P$,

(iv) $N_1P \neq N_2P$, *even* $N_2P \nsubseteq DEC$.

So by (ii) *for* $i > i'$ *we have* $DTIME(n^{i'}) \subset DTIME(n^i)$, $N_1TIME(n^{i'}) \subset N_1TIME(n^i)$, *and* $N_2TIME(n^{i'}) \subset N_2TIME(n^i)$.

PROOF. The universe of $\mathscr{A}$ is $\mathbf{N}$. Let $\mathscr{A}$ contain S, the addition, 2^n, and a bijection $\pi'\colon \mathbf{N} \times \mathbf{N} \to \mathbf{N}$ as functions, and 0 as a constant. Besides the corresponding symbols, the language $\mathscr{L}$ for $\mathscr{A}$ contains four binary relation symbols W, X, Y, and Z.

Let $(z_h)_{h \geq 0}$ be an enumeration of

$$\{(n, m, W)\colon n, m \in \mathbf{N}\} \cup \{(n, m, X)\colon n, m \in \mathbf{N}\} \cup \{(n, m, Z)\colon n, m \in \mathbf{N}\}.$$

By Lemma 1, 0, S, and π' already determine a bijection $\pi\colon \mathbf{N}^+ \to \mathbf{N}$, which will be computable in linear time over $\mathscr{A}$.

Let $Y^{\mathscr{A}} = \{(n, m)\colon m = f(n)\}$.

We next describe the idea of the proof. We will give a simultaneous construction of $W^{\mathscr{A}}$, $X^{\mathscr{A}}$, and $Z^{\mathscr{A}}$ in stages s. Similarly as in the previous sections we want to do this by fixing for more and more pairs (n, m) whether $(n, m) \in W^{\mathscr{A}}$, $(n, m) \in X^{\mathscr{A}}$, or $(n, m) \in Z^{\mathscr{A}}$ shall hold, respectively. This time however, we have to modify this approach as we will see below.

The idea to achieve $DTIME(n^i) \not\subseteq N_2TIME(n^{i'})$ for $i > i'$ is as follows. Given i, i' with $i > i'$, the set $\tilde{L}$ is defined for all inputs w by

$$w \in \tilde{L} \Leftrightarrow \{t : 0 \leq t \leq |w|^i \wedge (\pi(w), t) \in Z^{\mathscr{A}}\} = \emptyset.$$

Then $\tilde{L} \in DTIME(n^i)$. To get $\tilde{L} \notin N_2TIME(n^{i'})$, we will ensure $\tilde{L} \neq L(\tilde{N})$, for all $\mathscr{L}$-programs $\tilde{N}$ nondeterministic of the second kind, such that over all $\mathscr{L}$-structures each computation of $\tilde{N}$ on an input of length n has length $\leq t(n)$ for some $t \in O(n^{i'})$. Given such an $\tilde{N}$, at a point in the construction we choose an input w with $|w|^i \geq t(|w|)$ as a witness and ensure that $\{t : 0 \leq t \leq |w|^i \wedge (\pi(w), t) \in Z^{\mathscr{A}}\}$ is empty iff $\tilde{N}(w)$ does not seem to accept. However in a computation nondeterministic of the second kind we can guess arbitrary elements of the structure, so whether $\tilde{N}(w)$ will accept over $\mathscr{A}$ can depend on the whole structure. Hence it can happen that we first think that $\tilde{N}(w)$ does not accept and cause $\{t : 0 \leq t \leq |w|^i \wedge (\pi(w), t) \in Z^{\mathscr{A}}\}$ to be empty, but later it seems that $\tilde{N}(w)$ accepts and we change some definitions causing this set to become nonempty. Still later we might switch from nonempty to empty again and so on. But for each such $\tilde{N}(w)$ there will be only finitely many such changes and in the end our strategy succeeds, i.e., $\tilde{L} \neq L(\tilde{N})$.

So we change definitions in the construction, but for each pair (n, m) we switch from $(n, m) \in W^{\mathscr{A}}$, $(n, m) \in X^{\mathscr{A}}$, $(n, m) \in Z^{\mathscr{A}}$ to $(n, m) \notin W^{\mathscr{A}}$, $(n, m) \notin X^{\mathscr{A}}$, $(n, m) \notin Z^{\mathscr{A}}$ and vice versa only finitely many times, respectively. Hence we finally obtain a well-defined $\mathscr{A}$.

Additionally using the set $Y^{\mathscr{A}}$ our strategy to get $DTIME(n^i) \not\subseteq N_2TIME(n^{i'})$ for $i > i'$ also guarantees $DTIME(f) \not\subseteq N_2P$.

We want to achieve $P \neq N_1P$ over $\mathscr{A}$ similarly as in Theorem 3. We will arrange that $L \in N_1P \setminus P$, where L is defined as follows:

$$w \in L \Leftrightarrow \exists j < 2^{|w|}\ (\pi(w), j) \in W^{\mathscr{A}}.$$

The idea is to pick a witness w for each deterministic M with polynomially bounded computations and to ensure in the construction that $M(w)$ accepts iff $w \in L$. Since P is closed under complements by the remark after Lemma 2, for all $J \in P$ there will then be a w with $w \in J$ iff $w \notin L$.

We will build $L' \in N_2P \setminus DEC$, L' consisting of strings of length 1 with

$$(n) \in L' \Leftrightarrow \exists j\ (n, j) \in X^{\mathscr{A}}.$$

Now the idea is to pick a witness $n \in \mathbf{N}$ for each program N^* and to ensure in the construction that $(n) \in L'$ iff N^* accepts (n). Then $\mathbf{N}^+ \setminus L'$ cannot be recognized by any program, so $L' \notin DEC$. Since $N_1P \subseteq DEC$ we will also have $N_1P \neq N_2P$, similarly as in Theorem 3.

We will use a finite injury priority argument with requirements of the following kind:

(a) for all $i, i', i > i'$, for all $\tilde{N}$ as above the requirement that for some $w \in \mathbf{N}^+$, $\tilde{N}(w)$ accepts iff $w \notin \tilde{L}$,
(b) for all M as above the requirement that for some $w \in \mathbf{N}^+$, $M(w)$ accepts iff $w \in L$,
(c) for all N^* as above the requirement that for some $n \in \mathbf{N}$, $(n) \in L'$ iff $N^*(n)$ accepts.

Each requirement is injured only finitely many times by another requirement of higher priority and can thus be satisfied. In the important case such an injury consists of changing some definitions in order to satisfy the requirement of higher priority such as switching from $(n,m) \in Z^{\mathscr{A}}$ to $(n,m) \notin Z^{\mathscr{A}}$ for some (n,m) and thereby destroying the strategy for the requirement of lower priority. If a requirement is injured we restart with a new witness or change some definitions. We remark that if we always restart with a new witness in order to avoid changes of definitions it might happen that we have to restart infinitely many times and thus do not satisfy the requirement.

Hence we give a construction for $W^{\mathscr{A}}, X^{\mathscr{A}}, Z^{\mathscr{A}}$ in stages s performed in increasing order of s of the following kind: At the beginning of this construction for all z_i the corresponding evaluation, which can have value 0 or 1, is undefined. (By this we mean, that e.g., if $z_i = (n,m,Z)$, then it is not yet determined, whether $(n,m) \in Z^{\mathscr{A}}$ or whether $(n,m) \notin Z^{\mathscr{A}}$). The evaluation corresponding to z_i may then be defined at a point in the construction. Once defined, it is never "undefined", but may change from 0 to 1 or from 1 to 0. Given a point in the construction, if this evaluation is defined, we denote its current value by z_i^{cv}. If it is undefined, z_i^{cv} is undefined as well. For $z_i = (n,m,Z)$ we mean by $z_i^{cv} = 0$ that $(n,m) \notin Z^{\mathscr{A}}$ shall hold and $z_i^{cv} = 1$ means that $(n,m) \in Z^{\mathscr{A}}$ shall hold. We do similarly for (n,m,W) and (n,m,X). If at a point in the construction we set $z_i^{cv} = b$, $b \in \{0,1\}$, we mean that from that point on we have $z_i^{cv} = b$ unless this is redefined later. Then z_i^s refers to the value of z_i^{cv} at the end of stage s. Instead of $(n,m,Z)^{cv}$ we also write $Z^{cv}(n,m)$ and so on. We will see that for each z_i, there will be a point in the construction, such that z_i^{cv} is defined and does not change any more later, hence we will obtain a well-defined $\mathscr{A}$, as already remarked above.

Given a point in the construction, we will also have the notion of a computation of a program N on an input w. Such a computation is intended to be a computation over the structure $\mathscr{A}$ under construction. Since $W^{\mathscr{A}}$, $X^{\mathscr{A}}$, and $Z^{\mathscr{A}}$ are not yet completely specified during their construction, we have to say how to handle conditional instructions with *Cond* of the form $W(\tilde{p}_j, \tilde{p}_{j'})$, $X(\tilde{p}_j, \tilde{p}_{j'})$, or $Z(\tilde{p}_j, \tilde{p}_{j'})$. Suppose in the following n is the number in the position of the jth pointer and m the number in the position of the j'th pointer. If we want to perform such an instruction, then we require that $W^{cv}(n,m)$, $X^{cv}(n,m)$, or $Z^{cv}(n,m)$ is defined and that with *Cond* we associate $W^{cv}(n,m)$, $X^{cv}(n,m)$, or $Z^{cv}(n,m)$, respectively, in order to perform this instruction. Otherwise the computation is not allowed. For short we say that we *need* $W^{cv}(n,m)$, $X^{cv}(n,m)$, *or* $Z^{cv}(n,m)$ *for the computation*, respectively, if we perform such an instruction in such a computation. Hence we also can have accepting computations of $N^s(w)$, where we refer to the end of stage s. Due to later changes of definitions our plan that such computations will be actually computations over $\mathscr{A}$ might not always work, however.

Let $M_0, M_1, \dots$ be a sequence of deterministic $\mathscr{L}$-programs and $q'_0, q'_1, \dots$ a sequence of polynomials according to Lemma 2(i). Let $\tilde{N}_0, \tilde{N}_1, \dots$ be a sequence of $\mathscr{L}$-programs nondeterministic of the second kind and $\tilde{q}_0, \tilde{q}_1, \dots$ be a sequence of polynomials according to Lemma 2(iii). Further let $N_0^*, N_1^*, \dots$ be an enumeration of all $\mathscr{L}$-programs nondeterministic of the second kind.

To achieve (i) of the theorem we will satisfy for all i

R_{3i}: $\exists w \ \left(M_i(w) \text{ accepts over } \mathscr{A} \Leftrightarrow \exists j < 2^{|w|}\ (\pi(w), j) \in W^{\mathscr{A}}\right)$.

To get (ii) and (iii) we will satisfy for all i

R_{3i+1}: For the numbers j, n with $\pi'(j, n) = i$ there is a w of length $n+1$ such that

$$\tilde{N}_j(w) \text{ accepts over } \mathscr{A} \Leftrightarrow \left\{t\colon 0 \le t \le \tilde{q}_j(n+1) \wedge (\pi(w), t) \in Z^{\mathscr{A}}\right\} \neq \emptyset.$$

Moreover we require $(\pi(w), t) \notin Z^{\mathscr{A}}$ if $t > \tilde{q}_j(n+1)$ and if $\tilde{N}_j(w)$ does not accept over $\mathscr{A}$.

Finally, for (iv) we satisfy for all i

R_{3i+2}: $\exists n \in \mathbf{N}\ (N_i^* \text{ accepts } (n) \text{ over } \mathscr{A} \Leftrightarrow \exists j\ (n, j) \in X^{\mathscr{A}})$.

In order to satisfy some R_e we may fix some values z_h^{cv} during some stage of the construction and want that these z_h^{cv} are not changed later. So for all e we have a restraint set $U_e^s \subseteq \{z_h\colon h \ge 0\}$ defined at stage s. If z_h is put into U_e^s, then at stage $s+1$ the value z_h^{cv} can only be changed to satisfy some $R_{e'}$, $e' \le e$, namely to satisfy R_e or some requirement of higher priority. For all e and s, U_e^s will be finite.

Moreover, at stage $s+1$ we have a string w_e^s that is a potential witness to satisfy R_e. Here w_e^s is defined by induction on e as the string w with minimal $\pi(w)$ and $w \neq w_{e'}^s$ for $e' < e$, such that

(i) if $e = 3i$ we have for all k that $(\pi(w), k, W) \notin \bigcup_{e'<e} U_{e'}^s$ and $2^{|w|} > q'_i(|w|)$ hold,

(ii) if $e = 3i+1$ and $\pi'(j, n) = i$ we have for all k that $(\pi(w), k, Z) \notin \bigcup_{e'<e} U_{e'}^s$ and $|w| = n+1$ hold,

(iii) if $e = 3i+2$ we have $w = (n)$ for some $n \in \mathbf{N}$, namely we have $|w| = 1$, and for all k we have $(n, k, X) \notin \bigcup_{e'<e} U_{e'}^s$.
(Here $\bigcup_{e'<e} U_{e'}^s = \emptyset$ if $e = 0$).

We say R_{3i} *requires attention at stage* $s+1$ iff neither (1) nor (2) holds:

(1) For $w = w_{3i}^s$, $M_i^s(w)$ has an accepting computation, all z_h with z_h^s needed for this computation are in U_{3i}^s, for some $k < 2^{|w|}$ we have $W^s(\pi(w), k) = 1$.

(2) For $w = w_{3i}^s$, $M_i^s(w)$ has no accepting computation and for no $k < 2^{|w|}$ $W^s(\pi(w), k) = 1$.

R_{3i+1} *requires attention at stage* $s+1$ iff neither (1) nor (2) holds, where $\pi'(j, n) = i$:

(1) $\tilde{N}_j^s(w_{3i+1}^s)$ has an accepting computation p, such that all z_h with z_h^s needed for p are in U_{3i+1}^s, $\{t\colon 0 \le t \le \tilde{q}_j(n+1) \wedge Z^s(\pi(w_{3i+1}^s), t) = 1\} \neq \emptyset$.

(2) $\tilde{N}_j^s(w_{3i+1}^s)$ has no accepting computation, for all t we have $Z^s(\pi(w_{3i+1}^s), t) = 0$ if $Z^s(\pi(w_{3i+1}^s), t)$ is defined.

R_{3i+2} *requires attention at stage* $s+1$ iff neither (1) nor (2) holds, where $w_{3i+2}^s = (n)$:

(1) $N_i^{*s}(w_{3i+2}^s)$ has an accepting computation, such that all z_h with z_h^s needed for this computation are in U_{3i+2}^s. For some k we have $X^s(n,k) = 1$.

(2) $N_i^{*s}(w_{3i+2}^s)$ has no accepting computation and for no k we have $X^s(n,k) = 1$.

R_e receives attention at stage $s+1$, if e is minimal, such that R_e requires attention. At any stage $s+1$ some R_e will receive attention, e.g., since by construction of the sequence $(M_i)_{i\geq 0}$ there are infinitely many i, such that $M_i^s(w)$ has an accepting computation for all w (and such that we do not need any z_h^s for these computations). But $W^s(\pi(w),k) = 1$ will hold for only finitely many pairs (w,k). So some R_{3i} requires attention at stage $s+1$.

Now we can state the construction. Note that at any point there will be only finitely many z_h with z_h^{cv} defined, that for all s there will be only finitely many e with $U_e^s \neq \emptyset$, and that all U_e^s will be finite. Perform in increasing order of s the following stages.

Construction.

Stage 0. All z_h^{cv} are undefined, and for all e, $U_e^0 = \emptyset$.

Stage s+1. Let e be the index, such that R_e receives attention at stage $s+1$.

Step 1. (To satisfy R_e at stage $s+1$).

Case 1: $e = 3i$. For $w = w_e^s$, if $M_i^s(w)$ has an accepting computation set $W^{cv}(\pi(w),k) = 1$ for some $k < 2^{|w|}$, such that $W^s(\pi(w),k)$ is not needed for this accepting computation. If $M_i^s(w)$ has no accepting computation set $W^{cv}(\pi(w),k) = 0$ for all $k < 2^{|w|}$.

Case 2: $e = 3i+1$. Let j, n be the numbers such that $\pi'(j,n) = i$. Let $T = \{(\pi(w_e^s), t, Z) \colon 0 \leq t \leq \tilde{q}_j(n+1)\}$.

First suppose $\tilde{N}_j^s(w_e^s)$ has an accepting computation p. Fix such a p and let $W = T \setminus \{(l,m,Z) \colon Z^s(l,m) \text{ needed for } p\}$. Then $W \neq \emptyset$. Set $Z^{cv}(\pi(w_e^s),t) = 1$ for the least t with $(\pi(w_e^s),t,Z) \in W$.

Second suppose $\tilde{N}_j^s(w_e^s)$ has no accepting computation. Set $Z^{cv}(\pi(w_e^s),t) = 0$ if $Z^s(\pi(w_e^s),t)$ is defined.

Case 3: $e = 3i+2$. Then $w_e^s = (n)$ for some $n \in \mathbf{N}$. If $N_i^{*s}(w_e^s)$ has an accepting computation set $X^{cv}(n,k) = 1$ for some k, such that $X^{cv}(n,k)$ is not yet defined. If $N_i^{*s}(w_e^s)$ has no accepting computation, set $X^{cv}(n,k) = 0$ for all k, such that $X^s(n,k)$ is defined.

Step 2. (Defining the sets U_h^{s+1}). If $h = e$ let $U_h^{s+1} = \{z_l \colon z_l^{cv} \text{ defined now}\}$ and if $h \neq e$ let $U_h^{s+1} = U_h^s$.

Step 3. (To complete the definition of $W^{\mathscr{A}}$, $X^{\mathscr{A}}$, and $Z^{\mathscr{A}}$). Set $z_h^{cv} = 0$ for the least h such that z_h^{cv} is not yet defined. This finishes the construction.

First we check that the construction can be performed successfully. The next two lemmas are needed in order to see that this construction yields well-defined relations $W^{\mathscr{A}}$, $X^{\mathscr{A}}$, and $Z^{\mathscr{A}}$, and to prepare the proof that $\mathscr{A}$ has the desired properties.

Lemma 8. *For all e there is a number s_e, such that*:

(i) *R_e does not require attention at stage $t+1$, $t \geq s_e$.*

(ii) *$U_e^{s_e} = U_e^{t+1}$ for $t \geq s_e$. For $z_d \in U_e^{s_e}$ the value $z_d^{s_e}$ exists and z_d^{cv} does not change any more later.*

(iii) *$w_e^{s_e} = w_e^{t+1}$ for all $t \geq s_e$.*

PROOF. The proof is by induction on e. So fix e and assume the lemma to be true for $l < e$. Let q be a number, such that $q \geq s_0, \dots, s_{e-1}$, ($q$ arbitrary, if $e = 0$). By (ii) and (iii) of the induction hypotheses, $w_e^q = w_e^{t+1}$ for all $t \geq q$. Let $w = w_e^q$. If R_e does not require attention at stages $t + 1$ with $t \geq q$ we can let $s_e = q$. So suppose R_e requires and hence receives attention at some stage $u + 1$ with $u \geq q$.

Case 1: $e = 3i$. If $M_i^u(w)$ has an accepting computation, then by construction we can choose $s_e = u + 1$. If $M_i^u(w)$ has no accepting computation and if R_e does not require attention at stages $t + 1$ with $t > u$, then we can let $s_e = u + 1$. If $M_i^u(w)$ has no accepting computation and if R_e requires attention at a least stage $t + 1$ with $t > u$, then $M_i^t(w)$ must have an accepting computation. So we can let $s_e = t + 1$.

The argument for the cases where $e = 3i + 1$ or $e = 3i + 2$ is similar to that for case 1. ⊣

LEMMA 9. *For all z_h there is a point in the construction, such that z_h^{cv} is defined and does not change later.*

PROOF. Fix z_h. By step 3 of the construction z_h^{cv} will be defined at some point in the construction. By step 2, $z_h \in U_e^s$ for some e and s. So the conclusion follows from Lemma 8 since $U_e^s \subseteq U_e^{s_e}$. ⊣

By Lemma 9 we see that the construction yields a well-defined $\mathscr{A}$, because for all z_h we have a final $z_h^{cv} \in \{0, 1\}$, denoted by $z_h^{\mathscr{A}}$. It is clear what we mean by saying $z_h^{\mathscr{A}}$ is needed in some computation over $\mathscr{A}$.

LEMMA 10. *All R_e are satisfied.*

PROOF. Fix e. Let $w = w_e^{s_e}$, $w_e^{s_e}$ from Lemma 8.

Case 1: $e = 3i$. If $M_i(w)$ accepts over $\mathscr{A}$, then by Lemma 9 there is an $s' \geq s_e$, such that $M_i^{s'}(w)$ has an accepting computation. Namely choose $s' \geq s_e$, such that $z_h^{s'} = z_h^{\mathscr{A}}$ for all z_h with $z_h^{\mathscr{A}}$ needed for the computation of $M_i(w)$ over $\mathscr{A}$, (we remark, that we can even choose $s' = s_e$). Since R_e does not require attention at stage $s' + 1$ for some $k < 2^{|w|}$ we have $W^{s'}(\pi(w), k) = 1$. Since R_e does not receive attention at stage $s + 1$, $s \geq s'$, and since for all $e' \neq e$ and $s \geq s'$ we have $w_{e'}^s \neq w$, $W^{cv}(\pi(w), k) = 1$ always after stage s'. So $(\pi(w), k) \in W^{\mathscr{A}}$ and R_e is satisfied.

If $M_i(w)$ does not accept over $\mathscr{A}$, then $M_i^{s_e}(w)$ does not have an accepting computation for otherwise by (i) and (ii) of Lemma 8, $M_i(w)$ would accept over $\mathscr{A}$. By Lemma 8(i) there is no $k < 2^{|w|}$ with $W^{s_e}(\pi(w), k) = 1$. Since R_e does not receive attention at stage $s + 1$, $s \geq s_e$, and since for all $e' \neq e$ and $s \geq s_e$ we have $w_{e'}^s \neq w$, $W^{cv}(\pi(w), k) = 1$ holds for no $k < 2^{|w|}$ after stage s_e. So $(\pi(w), k) \in W^{\mathscr{A}}$ for no $k < 2^{|w|}$ and R_e is satisfied.

Case 2: $e = 3i + 1$. Let j, n be the numbers with $\pi'(j, n) = i$. If $\tilde{N}_j(w)$ accepts over $\mathscr{A}$ there is an $s' \geq s_e$ such that $\tilde{N}_j^{s'}(w)$ has an accepting computation. Since R_e does not require attention at stage $s' + 1$ we have

$$\left\{t \colon 0 \leq t \leq \tilde{q}_j(n+1) \wedge Z^{s'}(\pi(w), t) = 1\right\} \neq \emptyset$$

and similarly as above for all $0 \leq t \leq \tilde{q}_j(n + 1)$ the value $Z^{cv}(\pi(w), t)$ does not change after stage s' if $Z^{s'}(\pi(w), t)$ is defined. So

$$\left\{t \colon 0 \leq t \leq \tilde{q}_j(n+1) \wedge (\pi(w), t) \in Z^{\mathscr{A}}\right\} \neq \emptyset.$$

If $\tilde{N}_j(w)$ does not accept over $\mathscr{A}$, then $\tilde{N}_j^{s_e}(w)$ has no accepting computation, for otherwise by (i) and (ii) of Lemma 8, $\tilde{N}_j(w)$ would accept over $\mathscr{A}$. Since R_e does not require attention at stage $s_e + 1$,

$$\{t\colon Z^{s_e}(\pi(w), t) = 1\} = \emptyset,$$

and similarly as above there is no t, such that we set $Z^{cv}(\pi(w), t) = 1$ after stage s_e. So

$$\left\{t\colon (\pi(w), t) \in Z^{\mathscr{A}}\right\} = \emptyset.$$

Case 3: $e = 3i + 2$. This is again similar to case 1. ⊣

LEMMA 11. *$P \neq N_1P$ over $\mathscr{A}$.*

PROOF. Let L be defined by $w \in L$ iff $\exists j < 2^{|w|}$ $(\pi(w), j) \in W^{\mathscr{A}}$. Then $L \in N_1P$ as in the proof of Theorem 3. Since all R_{3i} are satisfied and by the remark after Lemma 2, for each set $J \in P$ there is a w and an i, such that $w \notin J$ iff $M_i(w)$ accepts over $\mathscr{A}$ iff $w \in L$. ⊣

LEMMA 12. *For i, i' with $i > i'$ we have $DTIME(n^i) \not\subseteq N_2TIME(n^{i'})$ over $\mathscr{A}$.*

PROOF. Fix i and i' with $i > i'$. Let the set $\tilde{L}$ of inputs be defined by

$$w \in \tilde{L} \Leftrightarrow \left\{t\colon 0 \leq t \leq |w|^i \wedge (\pi(w), t) \in Z^{\mathscr{A}}\right\} = \emptyset.$$

Then $\tilde{L} \in DTIME(n^i)$, note that with an idea from the proof of Lemma 2, $|w|^i$ can be computed in a tape cell using 0 and S. To see $\tilde{L} \notin N_2TIME(n^{i'})$ fix $\hat{L} \in N_2TIME(n^{i'})$. Choose j such that $\tilde{q}_j \in O(n^{i'})$ and $\hat{L} = L(\tilde{N}_j)$. Choose $n \geq 1$ such that $n^i \geq \tilde{q}_j(n)$. Since $R_{3\pi'(j,n-1)+1}$ is satisfied by Lemma 10, there is a w of length n such that

$$\tilde{N}_j(w) \text{ does not accept } \Leftrightarrow \left\{t\colon 0 \leq t \leq \tilde{q}_j(n) \wedge (\pi(w), t) \in Z^{\mathscr{A}}\right\} = \emptyset$$

and $(\pi(w), t) \notin Z^{\mathscr{A}}$ if $t > \tilde{q}_j(n)$ and $\tilde{N}_j(w)$ does not accept. So $\tilde{N}_j(w)$ does not accept iff $\{t\colon 0 \leq t \leq n^i \wedge (\pi(w), t) \in Z^{\mathscr{A}}\} = \emptyset$ iff $w \in \tilde{L}$. Hence $\tilde{L} \neq \hat{L}$ and $\tilde{L} \notin N_2TIME(n^{i'})$. ⊣

LEMMA 13. *$DTIME(f) \not\subseteq N_2P$ over $\mathscr{A}$.*

PROOF. Let L_f be the set of inputs defined by

$$w \in L_f \Leftrightarrow \left\{t\colon 0 \leq t \leq f(|w|) \wedge (\pi(w), t) \in Z^{\mathscr{A}}\right\} = \emptyset.$$

Then $L_f \in DTIME(f)$. Note that we can compute $|w|$ and then $f(|w|)$ in a cell of the input and work tape on input w using 0, S, and $Y^{\mathscr{A}}$. To see $L_f \notin N_2P$, we fix j and see similarly as in Lemma 12 that $L_f \neq L(\tilde{N}_j)$: Choose $n \geq 1$ such that $f(n) \geq \tilde{q}_j(n)$. Since $R_{3\pi'(j,n-1)+1}$ is satisfied by Lemma 10, there is a w of length n, such that $\tilde{N}_j(w)$ does not accept iff $\{t\colon 0 \leq t \leq \tilde{q}_j(n) \wedge (\pi(w), t) \in Z^{\mathscr{A}}\} = \emptyset$ and $(\pi(w), t) \notin Z^{\mathscr{A}}$ for $t > \tilde{q}_j(n)$ if $\tilde{N}_j(w)$ does not accept. So $\tilde{N}_j(w)$ does not accept iff $w \in L_f$. ⊣

LEMMA 14. *$N_2P \not\subseteq DEC$ over $\mathscr{A}$.*

PROOF. $L' \in N_2P$, L' consisting of words of length 1 with $(n) \in L'$ iff $\exists j$ $(n, j) \in X^{\mathscr{A}}$. Further, since all R_{3i+2} are satisfied, for all i there is an n, such that N_i^* accepts (n) over $\mathscr{A}$ iff $\exists j$ $(n, j) \in X^{\mathscr{A}}$ iff $(n) \notin \mathbf{N}^+ \setminus L'$. So $\mathbf{N}^+ \setminus L'$ is not recognizable at all and $L' \notin DEC$. ⊣

LEMMA 15. *$N_1P \neq N_2P$ over $\mathscr{A}$.*

PROOF. By Lemma 14 it suffices to see $N_1P \subseteq DEC$. Given $\hat{L} \in N_1P$ recognized by some N_i according to Lemma 2(ii), $\mathbf{N}^+ \setminus \hat{L}$ can be recognized as follows. On input w, deterministically generate all computations of $N_i(w)$, one after another, and accept iff no accepting one is found. ⊣

This concludes the proof of the theorem. ⊣

COROLLARY 2. *In Theorem 5 we can replace the function f by any countable family $(f_n)_{n\geq 0}$ of functions $f_n\colon \mathbf{N} \to \mathbf{N}$, such that each f_n dominates all polynomials. This means that for any such family $(f_n)_{n\geq 0}$ there is a structure $\mathscr{A}$ of finite signature with the following properties*:

(i) $P \neq N_1P$,
(ii) $DTIME(n^i) \not\subseteq N_2TIME(n^{i'})$ *if* $i > i'$,
(iii) *for all* n, $DTIME(f_n) \not\subseteq N_2P$,
(iv) $N_1P \neq N_2P$, *even* $N_2P \not\subseteq DEC$.

For example, we can choose as $(f_n)_{n\geq 0}$ the family of time constructible functions dominating all polynomials.

PROOF. Fix such a family $(f_n)_{n\geq 0}$ and let $(q_n)_{n\geq 0}$ be a sequence consisting exactly of all polynomials. Let $p(n,h) = max\{q_0(h),\dots,q_n(h)\}$ and define the function f by $f(h) = min\{f_n(h)\colon f_n(h) \geq p(n,h)\} \cup \{f_0(h)\}$. Then f dominates all polynomials, since given l, let k be a number such that for $n \leq l$ and $h \geq k$ we have $f_n(h) \geq q_l(h)$. Then for $h \geq k$, $f(h) \geq q_l(h)$. So we can apply Theorem 5. But $DTIME(f) \subseteq DTIME(f_n)$ for all n over any structure, since for all n, f_n dominates f. ⊣

On the other hand, for any structure $\mathscr{A}$ of finite signature there is some g dominating all polynomials, such that $DTIME(g) \subseteq P$, whence $DTIME(g) = P \subseteq N_2P$. To see this, fix such an $\mathscr{A}$. Suppose $M_0, M_1, \dots$ is a sequence of all deterministic programs M, such that $L(M) \notin P$ but $L(M) \in DTIME(f)$ witnessed by M for some f dominating all polynomials. Given i, define $f_i'(n) = 0$ if M_i does not accept any input of length n, $(f_i'(0) = 0)$. Otherwise, define $f_i'(n)$ to be the maximum t, such that M_i has an accepting computation of length t on an input of length n. Then $L(M_i) \in DTIME(f_i')$, so $f_i'(h) \geq q(h)$ for all polynomials q and infinitely many h. Again, let $(q_h)_{h\geq 0}$ be a sequence consisting exactly of all polynomials. Let $m_0 < m_1 < \dots$ be a sequence, such that $f_i'(m_j) \geq q_0(m_j) + \cdots + q_j(m_j)$. Let $m_{-1} = -1$ and let f_i be the function, such that for $j \geq 0$ and $m_{j-1} < h \leq m_j$ we have $f_i(h) = f_i'(m_j)$. Since polynomials (with natural numbers as coefficients) are nondecreasing, f_i dominates all polynomials. Let f be the function from the proof of the corollary obtained for the sequence $(f_i)_{i\geq 0}$. Let $g(h) = \lfloor\sqrt{f(h)}\rfloor$. Then g still dominates all polynomials. But $DTIME(g) \subseteq P$, since for all i there are infinitely many n such that $g(n)^2 \leq f(n) \leq f_i(n) = f_i'(n)$. By definition of $(f_i')_{i\geq 0}$, this implies $L(M_i) \notin DTIME(g)$ for all i and hence $DTIME(g) \subseteq P$.

We give some further remarks. In the following discussion, for $C \subseteq \mathbf{N}$ let U^C be the set of triples (e,n,t) classically recursive in C, such that the eth classical oracle Turing machine program in a fixed standard enumeration of all such (deterministic) programs halts in at most t steps on input $n \in \mathbf{N}$ with oracle C.

Of course, in Theorem 5 we can also prove that P consists of infinitely many different levels $DTIME(n^i)$ by the diagonalization technique used in the classical case. The corollary only for the case where $(f_n)_{n\geq 0}$ consists of all time constructible functions dominating all polynomials already follows from our strategy for (ii) of the theorem regardless how we choose f. Theorem 5 is maybe surprising, since on the one hand, (i), (ii), and (iv) state that N_2P contains many sets in some sense whereas (iii) states that N_2P contains only few sets in another sense. Note that $DTIME(f)$ is closed under complements in the proof of Theorem 5. There is also a proof of Theorem 5 using other ideas than our proof. As in Theorem 3 we have given a computability theoretic construction avoiding complications.

As proved in [6] over any structure of finite signature there are sets L p-m-complete as well for N_1P as for N_2P. However, since N_1P and N_2P consist of infinitely many different levels $N_1TIME(n^i)$ and $N_2TIME(n^i)$ over the structure from Theorem 5 we have for any such L over this structure, that the reductions cannot be computed in time $q(n)$ for some fixed polynomial q. We omit the obvious definitions of p-m-completeness and so on.

We can still prove further results like Theorem 4 or its corollary with similar proofs. For example, we can show that for some finite $\mathscr{L}$, the class of $\mathscr{L}$-structures with $N_2P = N_2TIME(n)$ is not closed under ultraproducts and not Δ-elementary. We construct a sequence $(\mathscr{A}_j)_{j\geq 0}$ of $\mathscr{L}$-structures with universes $\mathbf{N}$, such that in the construction of $\mathscr{A}_j$ we satisfy R_{3i+1} from the proof of Theorem 5 for $i \leq j$ using a finite injury priority argument. Then (ii) from Theorem 5 holds over $\Pi_D\mathscr{A}_j$, where D is as in the proof of Theorem 4. $N_2P = N_2TIME(n)$ is achieved over each $\mathscr{A}_j$ by ensuring for some appropriate constant c_j, some three-placed $V \in \mathscr{L}$ and all programs N that $N(w)$ accepts iff there are at least c_j numbers k with $(h, \pi(w), k) \in V^{\mathscr{A}_j}$. Here h depends on N and $\pi\colon \mathbf{N}^+ \to \mathbf{N}$ is a linear time computable bijection. Another way to achieve $N_2P = N_2TIME(n)$ is to add $U^{\emptyset}$ to $\mathscr{A}_j$ ahead of the construction to satisfy the named requirements and to ensure that all functions and relations of $\mathscr{A}_j$ become classically recursive, compare [11] Corollaries 1 and 2.

Fix $f\colon \mathbf{N} \to \mathbf{N}$ dominating all polynomials. Let $C = \{\pi'(n,m)\colon f(n) = m\}$, where $\pi'\colon \mathbf{N} \times \mathbf{N} \to \mathbf{N}$ is a classically recursive bijection. Using Shoenfield's Limit Lemma, we can easily ensure in the proof of Theorem 5 that $W^{\mathscr{A}}$, $X^{\mathscr{A}}$, and $Z^{\mathscr{A}}$ become elements of $\Delta_2^{AH,C}$, where $\Delta_2^{AH,C}$ refers to the classical arithmetical hierarchy AH relative to C. If we add U^C to $\mathscr{A}$ ahead of the construction of $W^{\mathscr{A}}$, $X^{\mathscr{A}}$, and $Z^{\mathscr{A}}$ and then construct $W^{\mathscr{A}}$, $X^{\mathscr{A}}$, and $Z^{\mathscr{A}}$, such that $W^{\mathscr{A}}, X^{\mathscr{A}}, Z^{\mathscr{A}} \in \Delta_2^{AH,C}$ we obtain a structure, which Theorem 5 holds for, such that the polynomial time hierarchy PH for nondeterminism of the second kind is proper. The latter can be concluded since the AH relative to C is proper. This PH is defined by $\Delta_0 = \Sigma_0 = \Pi_0 = P$ and $\Sigma_{k+1} = N_2P(\Sigma_k)$, $\Pi_{k+1} = co - N_2P(\Sigma_k)$, $\Delta_{k+1} = P(\Sigma_k)$, for more on this hierarchy see [11]. It is also possible to modify the construction of $\mathscr{A}$ using our methods so that we can show the properness of that PH by diagonalizing against polynomially bounded $\mathscr{L}$-oracle programs instead of referring to the AH, compare again [11].

As in Proposition 1, Theorem 5 also holds for any signature larger than the one we proved it for.

Acknowledgements. I thank the logic and computer science group of Greifswald for fruitful discussions. Special thanks go to Gunter Bär for supporting the writing of this paper. I am grateful to the anonymous referee for useful suggestions.

REFERENCES

[1] T. BAKER, J. GILL, and R. SOLOVAY, *Relativizations of the $P = NP$ question*, ***SIAM Journal on Computing***, vol. 4 (1975), pp. 431–442.

[2] J. L. BALCAZAR, J. DIAZ, and J. GABARRO, ***Structural complexity I, 2***, Springer, Berlin et al., 1988, 1990.

[3] J. L. BELL and A. B. SLOMSON, ***Models and ultraproducts***, North Holland, 1969.

[4] C. C. CHANG and H. J. KEISLER, ***Model theory***, North Holland, 1990.

[5] H. D. EBBINGHAUS, J. FLUM, and W. THOMAS, ***Mathematical logic***, Springer, 1984.

[6] A. HEMMERLING, *Computability and complexity over structures of finite type*, Preprint, 1995, Universität Greifswald, Reihe Mathematik, Nr. 2.

[7] ———, *Computability of string functions over algebraic structures*, ***Mathematical Logic Quarterly***, vol. 44 (1998), no. 1, pp. 1–44.

[8] ———, *Computability over structures of infinite signature*, ***Mathematical Logic Quarterly***, vol. 44 (1998), no. 3, pp. 394–416.

[9] ———, *On P versus NP for parameter-free programs over algebraic structures*, ***Mathematical Logic Quarterly***, vol. 47 (2001), no. 1, pp. 67–92.

[10] P. KOIRAN, *Computing over the reals with addition and order*, ***Theoretical Computer Science***, vol. 133 (1994), no. 1, pp. 35–48.

[11] G. MAINHARDT, *On structures with proper polynomial time hierarchy*, Preprint, 2002, Reihe Mathematik, Universität Greifswald, `http://sun-10.math-inf.uni-greifswald.de`.

[12] M. PRUNESCU, *A model-theoretic proof for $P \neq NP$ over all infinite abelian groups*, this JOURNAL, vol. 67 (2002), no. 1, pp. 235–238.

[13] R. I. SOARE, ***Recursively enumerable sets and degrees***, Springer, 1987.

UNIVERSITÄT GREIFSWALD
F.-L.-JAHNSTR. 15A
17487 GREIFSWALD, GERMANY
E-mail: gm@mail.uni-greifswald.de