# STRENGE SEPARATION VON *NP* UND *P* UNTER RELATIVIERUNG BEI GLEICHZEITIGEM KOLLAPS VON NICHTDETERMINISTISCHER ZU DETERMINISTISCHER EXPONENTIALZEIT

GERHARD LISCHKE
Sektion Mathematik der Friedrich-Schiller-Universität Jena
DDR – 6900 Jena, Universitätshochhaus

*(Eingegangen am 2. September 1985)*

## 1. Einführung

Eine der zentralen Fragestellungen in der Theorie der Kompliziertheit von Berechnungsprozessen wie in der theoretischen Computerwissenschaft überhaupt ist die Frage nach dem Verhältnis von Zeitklassen für deterministische und nichtdeterministische Berechnungsmodelle. Dabei ist insbesondere die Frage nach dem Verhältnis von deterministischen zu nichtdeterministischen Polynomialzeitklassen für Turing-Maschinen von praktischer Relevanz (vergl. [5]). Es ist trivial, daß jede durch eine deterministische Turing-Maschine in polynomialer Zeit akzeptierbare Menge auch durch eine nichtdeterministische Turing-Maschine in polynomialer Zeit akzeptierbar ist (man verwende nur dieselbe deterministische Maschine und fasse sie als nichtdeterministisches Modell auf). Die Frage nach der Umkehrung dieser Beziehung ist bis heute ungelöst und stellt das berühmte **P – NP**-Problem dar. Es zeigte sich nun, daß bei Relativierung der Fragestellung eine Beantwortung möglich ist. Anstelle der gewöhnlichen Turing-Maschinen werden sogenannte Orakel-Turing-Maschinen betrachtet, die den ersteren formal weitgehend ähnlich sind. Baker, Gill und Solovay [1] und unabhängig davon andere Autoren zeigten, daß das **P – NP**-Problem bei gewissen Relativierungen positiv, bei anderen dagegen negativ zu beantworten ist. Bezeichnen wir mit $\mathbf{P}^A$ bzw. $\mathbf{NP}^A$ die Klasse aller Mengen, die durch deterministische bzw. nichtdeterministische Orakel-Turing-Maschinen unter Benutzung der Orakelmenge $A$ akzeptiert werden können, so gibt es also Mengen $A$ mit $\mathbf{P}^A = \mathbf{NP}^A$ und andererseits Mengen $A$ mit $\mathbf{P}^A \neq \mathbf{NP}^A$. Betrachten wir nun zusätzlich als höhere Kompliziertheitsklassen die Exponentialzeitklassen **EL** und **NEL** (Definition siehe im Abschnitt 2), so ist bekannt, daß $\mathbf{P}^A = \mathbf{NP}^A$ auch $\mathbf{EL}^A = \mathbf{NEL}^A$ zur Folge hat (siehe z.B. [11]). Die relativierte Separation von **NP** und **P**, $\mathbf{P}^A \neq \mathbf{NP}^A$, ist jedoch von der Beziehung zwischen $\mathbf{EL}^A$ und $\mathbf{NEL}^A$ unabhängig. Es ist verhältnismäßig leicht, eine Menge $A$ zu finden, so daß $\mathbf{P}^A \neq \mathbf{NP}^A$ und $\mathbf{EL}^A \neq \mathbf{NEL}^A$ (siehe z.B. [9]). Das Auffinden einer Menge $A$ mit $\mathbf{P}^A \neq \mathbf{NP}^A$ und $\mathbf{EL}^A = \mathbf{NEL}^A$ ist hingegen mit einigen Schwierigkeiten verbunden, die im Abschnitt 3 diskutiert werden. Schließ-

lich präsentieren wir im Abschnitt 4 die Konstruktion einer Menge $A$, so daß die Klassen $\mathbf{P}^A$ und $\mathbf{NP}^A$ in gewissem Sinne streng separiert sind und die Exponentialzeitklassen $\mathbf{EL}^A$ und $\mathbf{NEL}^A$ zusammenfallen.

## 2. Das Berechnungsmodell und einige Bezeichnungen

Wir betrachten nur Mengen von Wörtern über dem Alphabet $\{0, 1\}$ und Maschinen mit diesem Alphabet als Eingabe- und Ausgabealphabet. $|x|$ bezeichne die Länge eines Wortes $x$. Eine Orakel-Turing-Maschine oder kurz Orakelmaschine ist eine Erweiterung einer mehrbändrigen Turing-Maschine, wie sie in [7] und [4] beschrieben wurde. Sie besteht aus einer Kontrolleinheit mit endlich vielen Zuständen, einem Nur-Lese-Eingabe-Band und einer endlichen Anzahl von Arbeitsbändern. Unter den Arbeitsbändern befindet sich als ausgezeichnetes Band das sogenannte Frage-Band. Unter den Zuständen befinden sich als ausgezeichnete Zustände ein Frage-Zustand, ein Yes-Zustand, ein No-Zustand und akzeptierende Zustände. Außerdem haben wir eine beliebige Menge $A \subseteq \{0, 1\}^*$ als sogenanntes Orakel oder Orakelmenge. Die Arbeitsweise einer Orakelmaschine ist nun zunächst genau dieselbe wie die einer gewöhnlichen Turing-Maschine mit folgenden Ergänzungen: Befindet sich die Maschine in einem gewissen Takt im Frage-Zustand, so geht sie in den Yes-Zustand über, falls das sich im entsprechenden Takt auf dem Frage-Band befindliche Wort (wir sagen, dieses Wort kommt als Orakelfrage vor) zu $A$ gehört, anderenfalls geht sie in den No-Zustand über. Im gleichen Takt wird noch der Inhalt des Frage-Bandes gelöscht und dann zum nächsten Takt übergegangen. Gemäß dieser verbalen Beschreibung ist eine formale Beschreibung leicht möglich, sei aber hier weggelassen. Wie üblich sprechen wir von deterministischen bzw. nichtdeterministischen Maschinen je nachdem, ob die unmittelbare Folgekonfiguration in jedem Takt eindeutig bestimmt ist oder nicht.

Ein Wort $x \in \{0, 1\}^*$ heißt von einer (deterministischen oder nichtdeterministischen) Maschine $\mathfrak{W}$ akzeptiert, wenn es eine Berechnung von $\mathfrak{W}$ mit der Eingabe $x$ gibt, die in einem akzeptierenden Zustand terminiert. Eine entsprechende Berechnung heiße akzeptierende Berechnung auf $x$. $\mathrm{Acc}(\mathfrak{W})$ sei die Menge aller von $\mathfrak{W}$ akzeptierten Wörter. Falls wir die Maschine $\mathfrak{W}$ mit dem Orakel $A$ betrachten, so schreiben wir $\mathfrak{W}^A$.

Ist $t$ eine Funktion von $\mathbf{N}$ in $\mathbf{N}$ und $\mathfrak{W}$ eine Orakelmaschine, so definieren wir: $\mathfrak{W}$ läuft in der Zeit $t$ genau dann, wenn für alle $n \in \mathbf{N}$ und alle Wörter $x \in \{0, 1\}^*$ mit der Länge $n$ jeder Berechnungsweg von $\mathfrak{W}^A$ bei beliebigem $A$ mit der Eingabe $x$ innerhalb von $t(n)$ Schritten terminiert.

$\mathbf{DTIME}^A(t)$ bzw. $\mathbf{NTIME}^A(t)$ sei die Klasse aller Mengen, die von deterministischen bzw. nichtdeterministischen Orakelmaschinen mit Orakel $A$ akzeptiert werden, die in der Zeit $t$ laufen. Als spezielle Klassen betrachten wir

$$\mathbf{P}^A =_{Df} \bigcup_{\substack{p \text{ ist ein} \\ \text{Polynom}}} \mathbf{DTIME}^A(p), \quad \mathbf{NP}^A =_{Df} \bigcup_{\substack{p \text{ ist ein} \\ \text{Polynom}}} \mathbf{NTIME}^A(p),$$

$$\mathbf{EL}^A =_{Df} \bigcup_{f \text{ ist linear}} \mathbf{DTME}^A(2^f),\ \mathbf{NEL}^A =_{Df} \bigcup_{f \text{ ist linear}} \mathbf{NTIME}^A(2^f).$$

**P**, **NP**, **EL** bzw. **NEL** bezeichnen die entsprechenden Klassen bezüglich gewöhnlicher Turing-Maschinen (unrelativierte Klassen). Wie in [1] seien $\{P_1, P_2, \ldots\}$ und $\{NP_1, NP_2, \ldots\}$ Numerierungen aller deterministischen bzw. nichtdeterministischen Orakelmaschinen, die in polynomialer Zeit laufen. O.B.d.A. nehmen wir an, daß $p_i(n)$ eine echte obere Schranke für die Länge aller Berechnungen von $P_i$ mit beliebigem Orakel auf einer Eingabe der Länge $n$ sei, wobei $p_i(n) = i + n^i$. $\{\mathrm{NEL}_1, \mathrm{NEL}_2, \ldots\}$ sei eine Numerierung aller nichtdeterministischen Orakelmaschinen, die in einer Zeit $2^f$ laufen, wobei $f$ eine lineare Funktion ist. Der Zusatz $^A$ an der Bezeichnung einer Maschine bedeute, daß wir diese Machine mit dem speziellen Orakel $A$ benutzen.

Die folgenden Abkürzungen werden in den anschließenden Konstruktionen benutzt.

Für natürliche Zahlen $i$ und $l$ und $x \in \{0, 1\}^*$ sei $\mathrm{cod}(i, x, l)$ das Wort

$$\underbrace{00\ldots0}_{i}1x\underbrace{00\ldots0}_{l} = 0^i1x10^l.$$

Für $T \in \{\mathrm{P}, \mathrm{EL}\}$, $A \subseteq \{0, 1\}^*$ und $n \in \mathbf{N}$ sei $\mathfrak{C}(\mathrm{NT}^A, n) =_{Df} \{\mathrm{cod}(i, x, l)$: $|\mathrm{cod}(i, x, l)| \leq n \wedge \mathrm{NT}_i^A$ akzeptiert $x$ in weniger als $l$ Schritten$\}$.
Mit diesen Bezeichnungen ist nun leicht die von Baker, Gill und Solovay in [1] angegebene Konstruktion einer Menge $A$ mit $\mathbf{P}^A = \mathbf{NP}^A$ zu beschreiben: Die Konstruktion erfolgt schrittweise, beginnend mit $A = \emptyset$, und im Schritt $n$ ist

$$A \leftarrow A \cup \mathfrak{C}(\mathrm{NP}^A, n)$$

zu setzen. Das bedeutet im Sinne einer Ergibtanweisung, daß die neu konstruierte Menge $A$ gleich dem rechts von $\leftarrow$ stehenden Wert mit der bisher definierten Menge $A$ zu setzen ist. Im Limes ergibt sich eine Menge $A$ mit $L \leqq_m^p A$ für jedes $L \in \mathbf{NP}^A$, was gleichbedeutend mit $\mathbf{NP}^A \subseteq \mathbf{P}^A$ ist. Denn ist $L$ durch eine nichtdeterministische in polynomialer Zeit laufende Orakelmaschine mit Orakel $A$ akzeptiert, so gilt $x \in L \leftrightarrow \mathrm{cod}(i, x, p(|x|)) \in A$ für ein gewisses festes $i \in \mathbf{N}$ und Polynom $p$. Die Funktion $f$ mit $f(x) = \mathrm{cod}(i, x, p(|x|))$ ist deterministisch in Polynomialzeit berechenbar. Die angegebene Konstruktion ist korrekt, denn um festzustellen, ob ein Wort $z = \mathrm{cod}(i, x, l)$ im Schritt $n$ zu $A$ hinzuzunehmen ist, braucht nur die Maschine $\mathrm{NP}_i^A$ auf $x$ für weniger als $l < n$ Schritte simuliert zu werden. Dabei könne nnur Fragen nach Wörtern mit Längen kleiner als $n$ vorkommen, und für diese Wörter ist vollständig bekannt, ob sie zu $A$ gehören oder nicht.

### 3. $\mathbf{EL}^A = \mathbf{NEL}^A$ bei gleichzeitiger Separation $\mathbf{P}^A \neq \mathbf{NP}^A$

Eine Menge $A$, für die gleichzeitig $\mathbf{EL}^A = \mathbf{NEL}^A$ und $\mathbf{P}^A \neq \mathbf{NP}^A$ gilt, wird ebenfalls schrittweise konstruiert. Wir benutzen wieder das Symbol $\leftarrow$ im Sinne von Ergibtanweisungen und beginnen mit $A = \emptyset$. Um die Separation von $\mathbf{P}^A$ und $\mathbf{NP}^A$ zu garantieren, konstruieren wir gleichzeitig schrittweise

eine Menge $B \in \mathbf{NP}^A \setminus \mathbf{P}^A$. Die Konstruktionen in [1] sowie in anderen Arbeiten legen zunächst die folgende Idee für den Schritt $n$ der Konstruktion nahe:

(1) $$A \leftarrow A \cup \mathfrak{C}(\mathrm{NEL}^A, 2^n).$$

Damit ist bereits analog zu oben der Kollaps von $\mathbf{NEL}^A$ zu $\mathbf{EL}^A$ garantiert ($x \in L \leftrightarrow \mathrm{cod}(i, x, 2^{f(|x|)}) \in A$ für geeignetes $i$ und lineare Funktion $f$ bei beliebigem $L \in \mathbf{NEL}^A$.). Nun ist noch über die deterministischen polynomialzeitbeschränkten Maschinen zu diagonalisieren. Dazu beginnen wir die Konstruktion mit $i = 1$ und $B = \emptyset$ und stellen nach Ausführung der Anweisung (1) die Frage, ob $2^n > p_i(n)$. Ist das nicht der Fall, so gehen wir zum nächsten Schritt $n+1$. Anderenfalls fragen wir nun, ob $0^n$ von $P_i^A$ akzeptiert wird. Wegen $2^n > p_i(n)$ kann dabei kein Codewort als Orakelfrage auftreten, das eventuell erst später in $A$ aufgenommen wird. Ist $0^n \in \mathrm{Acc}(P_i^A)$, so erhöhen wir $i \leftarrow i+1$ und gehen zum nächsten Schritt $n+1$. Anderenfalls setzen wir

(2) $$B \leftarrow B \cup \{0^n\},$$

(3) $A \leftarrow A \cup \{1z\colon |z| = q(n) \wedge 1z$ tritt nicht als Orakelfrage während der Arbeit von $P_i^A$ auf $0^n$ auf$\}$,

$i \leftarrow i+1$ und gehen zum nächsten Schritt $n+1$. $q$ ist dabei ein festes Polynom. Dadurch wird $B \neq \mathrm{Acc}(P_i^A)$ erreicht. Da nur weniger als $p_i(n) < 2^n$ Orakelfragen gestellt werden können, und es andererseits $2^{q(n)} \geq 2^n$ Wörter der Länge $q(n)$ gibt, existieren entsprechende Wörter $1z$. $q$ darf nicht größer als ein Polynom sein, weil sonst $B \in \mathbf{NP}^A$ nicht realisiert werden kann.

Die in (3) eventuell in $A$ aufgenommenen Wörter $1z$ haben aber für hinreichend großes $n$ eine Länge kleiner als $2^n$ und können somit den Teilschritt (1) von Schritt $n$ oder früheren Schritten beeinflussen. Damit ist unsere Konstruktion nicht brauchbar, und wir müssen nach Auswegen suchen. Nehmen wir in (3) nur solche Wörter $1z$ in $A$ auf, die weder bei der Arbeit von $P_i^A$ auf $0^n$ noch innerhalb von $l$ Schritten der Arbeit von $\mathrm{NEL}_j^A$ auf $x$ mit $|\mathrm{cod}(j, x, l)| \leq 2^n$ als Orakelfragen auftreten, so können diese zwar die Teilschritte (1) nicht beeinflussen, es kann aber sein, daß es gar keine derartigen Wörter gibt. Bei einer nichtdeterministischen Maschine können nämlich innerhalb von $l$ Schritten alle Wörter mit einer Länge kleiner als $c \cdot l$ mit konstantem Faktor $c < 1$ als Orakelfragen vorkommen. Da hier $l$ größenordnungsmäßig $2^n$ erreicht, müssen möglicherweise für hinreichend großes $n$ in (3) alle Wörter mit fester polynomialer Länge ausgeschlossen werden.

Als nächster Ausweg bietet sich die folgende Idee an. Jede Berechnung einer nichtdeterministischen Maschine ist eine Folge von Konfigurationen. Jede dieser Folgen läßt sich in gewisser Weise codieren, und die codierten Folgen lassen sich kanonisch ordnen. Man kann also beispielsweise von der ersten akzeptierenden Berechnung einer Maschine gemäß dieser kanonischen Ordnung sprechen (falls sie existiert). Wir definieren nun:

$$\mathrm{Qacc}(\mathrm{NEL}_j^A, x) =_{Df} \begin{cases} \{z\colon z \text{ kommt als Orakelfrage während der ersten akzeptierenden Berechnung von } \mathrm{NEL}_j^A \text{ auf } x \text{ vor}\} & \text{falls } x \in \mathrm{Acc}(\mathrm{NEL}_j^A) \\ \emptyset \text{ sonst.} \end{cases}$$

Ferner sei $Q(P_i^A, x)$ die Menge aller Wörter, die während der Arbeit von $P_i^A$ mit Eingabe $x$ als Orakelfragen vorkommen.
Wir verwenden nun eine andere Codierung, nämlich

$$\operatorname{cod}(i, x, l) = {}_{Df}\, 0^i 1 0^{2^{|x|}} 1 x 1 0^l$$

und ersetzen (3) durch

(3′) $$A \leftarrow A \cup \{1z\colon |z| = 3n+1 \wedge$$
$$1z \notin Q(P_i^A, 0^n) \cup \bigcup \{\operatorname{Qacc}(\mathrm{NEL}_j^A, x)\colon \exists l \operatorname{cod}(j, x, l) \in A\}\}.$$

Die Menge $Q(P_i^A, 0^n) \cup \bigcup \{\operatorname{Qacc}(\mathrm{NEL}_j^A, x)\colon \exists l \operatorname{cod}(j, x, l) \in A\}$ enthält wegen $p_i(n) < 2^n$ und $|\operatorname{cod}(j, x, l)| \leq 2^n$ weniger als $2^n + 2^{3n} \leq 2^{3n+1}$ Wörter. Es gibt also stets noch Wörter $z$ der Länge $3n+1$, so daß $1z$ nicht zu dieser Menge gehört. Diese Wörter können gemäß ihrer Bildung auch nicht die bisher in $A$ aufgenommenen Codewörter beeinflussen. Nun könnte es aber sein, daß derartige später in $A$ gelangende Wörter $1z$ bewirken, daß gewisse in früheren Schritten nicht in $A$ aufgenommene Codewörter doch zu $A$ gehören müßten. Damit ist auch dieser Ausweg zunächst nicht gangbar.

An dem Gesagten ändert sich nichts, wenn wir in (3) bzw. (3′) jeweils nur ein Wort $1z$ (das bezüglich der kanonischen Ordnung kleinste mit der entsprechenden Eingenschaft) in $A$ aufnehmen. Würden wir nach Aufnahme eines Wortes $1z$ in $A$ eine Neuberechnung von $A$ gemäß (1) durchführen, so könnte dann für das modifizierte $A$ $0^n \in \operatorname{Acc}(P_i^A)$ gelten, weshalb dann $1z$ gar nicht in $A$ sein dürfte.

Auch die Aufnahme kürzerer Codewörter zu $A$ in (1) liefert keine Lösung, da dann bei der Arbeit von $P_i^A$ auf $0^n$ nach Codewörtern gefragt werden könnte, die erst später in $A$ aufgenommen werden müssen. Die Wahl einer anderen Codierung und verschiedene Verfeinerungen der aufgezeigten Wege führen ebenfalls nicht zum Ziel. Es zeigt sich schließlich, daß der im Zusammenhang mit (3′) angegangene Weg erfolgreich ist, das prinzipielle Problem jedoch im Teilschritt (2) liegt.

In [6] bewies Hartmanis einen Satz, der in unserer Terminologie folgendermaßen zu formulieren ist:

**Satz** (Hartmanis [6]). $\mathbf{EL} = \mathbf{NEL}$ *gilt genau dann, wenn es keine dünne Menge in* $\mathbf{NP} \backslash \mathbf{P}$ *gibt.*

Eine Menge $S$ heißt dünn, wenn es ein Polynom $p_S$ gibt, so daß für jedes $n$ weniger als $p_S(n)$ Wörter mit maximaler Länge $n$ zu $S$ gehören.

Der Satz von Hartmanis gilt auch bei beliebigen Relativierungen: $\mathbf{EL}^A = \mathbf{NEL}^A$ genau dann, wenn keine dünne Menge in $\mathbf{NP}^A \backslash \mathbf{P}^A$ liegt.

Die von uns gemäß (2) konstruierte Menge $B$ ist offensichtlich dünn und kann demzufolge nicht die gewünschte Bedingung erfüllen. Eine erste Lösung des Problems präsentierte Wilson in [12] und [3], indem er eine Menge $A$ konstruierte, so daß $\mathbf{P}^A \neq \mathbf{NP}^A$ und $\mathbf{EL}^A = \mathbf{NEL}^A$. Kurtz [8] bewies die Existenz von Orakelmengen $A$ mit $\mathbf{P}^A \neq \mathbf{NP}^A$ und $\mathbf{NP}^A \backslash \mathbf{P}^A$ enthält keine dünne Menge. Gemäß des Satzes von Hartmanis sind beide Resultate äquivalent. Unabhängig davon wurde auch in [9] eine Menge $A$ mit diesen Eigenschaften konstruiert.

In [2, 10] und anderen Arbeiten wird eine schärfere Form der Separation relativierter Kompliziertheitsklassen betrachtet. Zwei Klassen $\mathfrak{A}$ und $\mathfrak{B}$ (mit $\mathfrak{A} \subseteq \mathfrak{B}$) heißen streng separiert, wenn es eine unendliche Menge $L$ in $\mathfrak{B} \setminus \mathfrak{A}$ gibt und keine unendliche Teilmenge von $L$ in $\mathfrak{A}$ liegt. $L$ heißt dann auch $\mathfrak{A}$-immun.

Es erhebt sich nun dei Frage, ob unter der Voraussetzung $\mathbf{EL}^A = \mathbf{NEL}^A$ eine strenge Separation von $\mathbf{P}^A$ und $\mathbf{NP}^A$ möglich ist. Äquivalent dazu ist die Frage, ob es Mengen $A$ gibt, so daß $\mathbf{NP}^A \setminus \mathbf{P}^A$ keine dünnen, aber $\mathbf{P}^A$-immune Mengen enthält. Diese Fragen sind positiv zu beantworten, und wir präsentieren im nächsten Abschnitt die Konstruktion entsprechender Mengen. Dabei werden die oben angedeuteten Versuche in verfeinerter Form ausgenutzt.

## 4. $\mathbf{EL}^A = \mathbf{NEL}^A$ bei strenger Separation von $\mathbf{P}^A$ und $\mathbf{NP}^A$

**Satz.** *Es gibt rekursive Mengen $A$ und $B$ mit den Eigenschaften* $\mathbf{EL}^A = \mathbf{NEL}^A$, $\mathbf{NP}^A \setminus \mathbf{P}^A$ *enthält keine dünne Menge,* $B \in \mathbf{NP}^A \setminus \mathbf{P}^A$ *und $B$ ist* $\mathbf{P}^A$*-immun.*

**Beweis.** Zur Vorbereitung der Konstruktion entsprechender Mengen wollen wir einige weitere Abkürzungen und Bezeichnungen vereinbaren. Anstelle der bisherigen Codierungen verwenden wir ab jetzt die folgende:

$$\operatorname{cod}(i, x, l) = {}_{Df} 0^{2^i} 1 0^{2^{4 \cdot |x|}} 1 x 1 0^{l^4}.$$

Mit dieser Codierung sei dann $\mathfrak{C}(\mathrm{NEL}^A, n)$ wie in **2.** definiert. Ferner sei:
$\mathfrak{C}^+(\mathrm{NEL}^A, n) = {}_{Df}\{\operatorname{cod}(i, x, l') : \exists l(\operatorname{cod}(i, x, l) \in \mathfrak{C}(\mathrm{NEL}^A, n) \wedge l' \geq l)\}$ und
$\mathfrak{C}^-(\mathrm{NEL}^A, n) = {}_{Df}\{\operatorname{cod}(i, x, l) : |\operatorname{cod}(i, x, l)| \leq n \wedge \mathrm{NEL}_i^A$ akzeptiert $x$ nicht innerhalb von $l$ Schritten$\}$.
Sind $A$ und $M$ Teilmengen von $\{0, 1\}^*$, $x \in \{0,1\}^*$ und $j$, $l \in \mathbf{N}$, so sei $\mathrm{COMP}(\mathrm{NEL}_j, A, M, x, l)$ die Menge aller Berechnungen der Maschine $\mathrm{NEL}_j$ auf $x$ mit weniger als $l$ Schritten, wobei alle Orakelfragen nach Wörtern aus $M$ jeweils in beliebiger, untereinander verträglicher Weise beantwortet werden, und alle sonstigen Orakelfragen in korrekter Weise gemäß $A$. Wir definieren dann:

$$\operatorname{fiacc}(\mathrm{NEL}_j, A, M, x, l) = {}_{Df} \begin{cases} \text{die erste akzeptierende Berechnung in} \\ \mathrm{COMP}(\mathrm{NEL}_j, A, M, x, l) \text{ falls diese} \\ \text{existiert; das leere Wort sonst.} \end{cases}$$

Ist $\mathfrak{P}$ eine Berechnung, so sei $M_{\mathfrak{P}}^+$ die Menge aller Wörter aus $M$, die während der Berechnung $\mathfrak{P}$ als Orakelfragen vorkommen und mit *Ja* beantwortet werden, und $Q_{\mathfrak{P}}$ sei die Menge aller Wörter, die während $\mathfrak{P}$ als Orakelfragen vorkommen.

Die Kostruktion erfolgt nun schrittweise und beginnt mit $n = i = 1$, $A = B = B' = L' = \varnothing, L = \{1\}, m = 0, M = 1 \cdot \bigcup_{k=1}^{\infty} \{0, 1\}^{2k-1}$. $n$ ist die Nummer des gerade auszuführenden Schrittes. $A$ und $B$ werden im Limes die gesuchten Mengen sein, die unseren Satz erfüllen. Dabei werden manche

Wörter vor ihrer Aufnahme in $B$ zunächst in $B'$ genommen. $L$ ist eine Liste von Indizes deterministischer polynomialzeitbeschränkter Maschinen, über die im entsprechenden Schritt zu diagonalisieren ist. $L'$ ist eine Teilmenge von $L$. $i$ ist der größte Index einer Maschine, über die jeweils zu diagonalisieren ist. $m$ ist ein Schwellenwert für gewisse Teilschritte der Konstruktion. Eine besondere Rolle spielt die Menge $M$. Gewisse Elemente aus $M$ werden wir in $A$ aufnehmen, nämlich genau diejenigen, auf die die Menge $B \leq_T^{NP}$ reduzierbar ist. Dieser Anteil von $A$ ist zu allen übrigen Elementen von $A$, die alle mit 0 beginnen, disjunkt. Wir reservieren zunächst in $M$ immer diejenigen Wörter, die keinen Einfluß auf andere Teile der Konstruktion haben, und für die später noch entsprechende Wörter in $B$ aufgenommen werden können. Die in **3.** geschilderten Schwierigkeiten überwinden wir dadurch, daß wir erstens wie dort alle Elemente aus $\mathrm{Qacc}(\mathrm{NEL}_j^A, x)$ mit $\mathrm{cod}(j, x, l) \in A$ aus $M$ entfernen und zweitens diejenigen Wörter aus $M$, die bewirken können, daß noch nicht zu $A$ gehörende Codewörter doch in $A$ sein müßten, bereits vorzeitig in $A$ aufnehmen. Im zweiten Falle müssen wir dann natürlich auch entsprechende Wörter in $B$ übernehmen. Aus später einzusehenden Gründen nehmen wir diese aber zunächst in $B'$.
Damit ergibt sich folgende allgemeine Beschreibung für den Schritt $n$ unserer Konstruktion:

$$A \leftarrow A \cup \mathfrak{C}^+(\mathrm{NEL}^A, 2^n)$$

$$M \leftarrow M \setminus \bigcup_{\mathrm{cod}(j,x,l) \in \mathfrak{C}(\mathrm{NEL}^A, 2^n)} \mathrm{Qacc}(\mathrm{NEL}_j^A, x)$$

Gibt es Wörter $z \in \mathfrak{C}^-(\mathrm{NEL}^A, 2^n)$ mit $|z| > 2^{n-1}$?
*Nein:* Gehe zu $\mathbf{M}_3$.
*Ja:* Dann sei $z_1, z_2, \ldots, z_{N_n}$ die kanonische Folge aller dieser Wörter. Setze $k \leftarrow 1$ und gehe zu $\mathbf{M}_1$.

$\mathbf{M}_1$: Setze $\mathfrak{P} \leftarrow \mathrm{fiacc}(\mathrm{NEL}_{j_k}, A, M, x_k, l_k)$, wobei $z_k = \mathrm{cod}(j_k, x_k, l_k)$.
Ist $\mathfrak{P}$ das leere Wort?
*Ja:* Dann gehe zu $\mathbf{M}_2$.
*Nein:* Setze $A \leftarrow A \cup \{\mathrm{cod}(j_k, x_k, l): l \geq l_k\} \cup M_{\mathfrak{P}}^+$,
$B' \leftarrow B' \cup \{z: \exists w(1zw \in M_{\mathfrak{P}}^+ \wedge |w| = |z| - 1)\}$,
$M \leftarrow M \setminus Q_{\mathfrak{P}}$ und gehe zu $\mathbf{M}_2$.

$\mathbf{M}_2$: $k = N_n$?
*Ja:* Gehe zu $\mathbf{M}_3$.
*Nein:* $k \leftarrow k+1$.
$z_k \in A$?
*Ja:* Gehe zu $\mathbf{M}_2$.
*Nein:* Gehe zu $\mathbf{M}_1$.

Damit sind die oben geschilderten Vorbereitungen erfolgt. In den Teilschritten $\mathbf{M}_3 - \mathbf{M}_6$ sichern wir die $\mathbf{P}^A$-Immunität von $B$, und in $\mathbf{M}_7$ erfolgt die eigentliche Diagonalisierung.

$\mathbf{M}_3$: $B'' \leftarrow \left\{z: z \in B' \wedge |z| \leq \frac{n}{2}\right\}$. Ist $n > 7$ und $\sum_{\substack{j \in L \\ z \in B''}} p_j(|z|) < 2^{n-3}$? $(*)$

(Ist $L$ oder $B''$ leer, so habe die Summe den Wert 0.)

*Ja*: Setze $M \leftarrow M \setminus \bigcup_{\substack{j \in L \\ z \in B''}} Q(P_j^A, z)$, $L' \leftarrow L' \cup \left\{j: j \in L \wedge \exists z\left(z \in B'' \wedge z \in Acc\left(P_j^A\right)\right)\right\}$,

$j_0 \leftarrow 0$ und gehe zu $\mathbf{M}_4$.

Da $B$ $\mathbf{P}^A$-immun werden soll, darf keine Menge der Form $\mathrm{Acc}(P_j^A)$ eine unendliche Teilmenge von $B$ sein. $L'$ ist die Menge derjenigen Indizes $j$, für die zunächst die Gefahr besteht, daß $\mathrm{Acc}(P_j^A)$ eine unendliche Teilmenge von $B$ wird. Um das zu verhindern, suchen wir als nächstes nach dem kleinsten Index $j_0 \in L'$, für den es ein $z_0 \in \mathrm{Acc}(P_{j_0}^A) \setminus B$ gibt. Finden wir ein solches Paar $(j_0, z_0)$, so werden wir sichern, daß das Wort $z_0$ niemals ein Element von $B$ wird, und streichen den Index $j_0$ aus unseren Listen. Natürlich können wir dabei nur nach Wörtern $z_0$ mit einer beschränkten Länge suchen.

Ist $n$ nicht groß genug, um $(*)$ mit *Ja* zu beantworten, so führen wir ebenfalls diese Suche durch, lassen dann aber später die Menge $B''$ unberücksichtigt (d.h. formal $B'' = \emptyset$).

*Nein:* Setze $B'' \leftarrow \emptyset$, $j_0 \leftarrow 0$ und gehe zu $\mathbf{M}_4$.

$\mathbf{M}_4$: Gibt es ein $j > j_0$ in $L'$?

*Ja:* Setze $j_0 \leftarrow \min\{j: j > j_0 \wedge j \in L'\}$,
$z_0 \leftarrow$ kleinstes Wort $z$ (bezüglich der kanonischen Ordnung aller Wörter) mit $z \notin B \cup B'$
und gehe zu $\mathbf{M}_5$.

*Nein:* Falls $(*)$ mit *Ja* beantwortet wurde, so setze
$B \leftarrow B \cup B''$, $B' \leftarrow B' \setminus B''$ und gehe zu $\mathbf{M}_7$,
sonst setze $M \leftarrow M \setminus \{0, 1\}^{2n}$
und gehe zum nächsten Schritt $n+1$.

Vor dem Übergang zu einem neuen Schritt $n+1$ entfernen wir alle Wörter der Länge $2n$ aus $M$, da diese auf spätere Schritte keinen Einfluß mehr haben können (Bei der Diagonalisierung im Schritt $n$ nehmen wir höchstens ein Wort der Länge $n$ in $B$ und entsprechende Wörter der Länge $2n$ aus $M$ in $A$ auf.).

$\mathbf{M}_5$: Setze $A_0 \leftarrow A$, $n_0 \leftarrow n$ und lasse die Maschine $P_{j_0}^{A_0}$ auf der Eingabe $z_0$ solange arbeiten, bis einer der folgenden Fälle eintritt:

*a)* es wird eine Orakelfrage nach einem Wort $q$ der Form $\mathrm{cod}(j, x, l)$ mit $q \notin A_0$ und $|q| > 2^{n_0}$ gestellt,

*b)* Berechnung beendet und $z_0 \notin \mathrm{Acc}(P_{j_0}^{A_0})$,

*c)* Berechnung beendet und $z_0 \in \mathrm{Acc}(P_{j_0}^{A_0})$.

Fall *a)*: Gehe zu $\mathbf{M}_6$.

$\mathbf{M}_6$: $n_0 \leftarrow n_0 + 1$, $A_0 \leftarrow A_0 \cup \mathfrak{C}^{-1}(\mathrm{NEL}^{A_0}, 2^{n_0})$.

Ist $q \in A_0$ oder $2^{n_0} \geq |q|$?
*Nein:* Gehe zu $\mathbf{M}_6$.
*Ja:* Dann beantworte die Orakelfrage in der entsprechenden Weise und arbeite weiter bis einer der Fälle *a), b), c)* eintritt.

Fall *b):* Gibt es ein $z > z_0$ (bezügl. der kanonischen Ordnung)

$$\text{mit } z \notin B \cup B' \text{ und } |z| \leq 2^{\frac{n-3}{4}} ?$$

*Ja:* $z_0$ sei das bezügl. der kanonischen Ordnung kleinste derartige Wort, gehe zu $\mathbf{M}_5$.

*Nein:* Gehe zu $\mathbf{M}_4$.

Fall *c):* Ist $n_0 > n$, so setze $A \leftarrow A_0$, $n \leftarrow n_0$, $j_0 \leftarrow 0$ und gehe zu $\mathbf{M}_4$,
sonst setze $L' \leftarrow L' \setminus \{j_0\}$, $L \leftarrow L \setminus \{j_0\}$, $B \leftarrow B \cup B''$,
$B' \leftarrow B' \setminus B''$, $m \leftarrow \max\{m, p_{j_0}(|z_0|), 2 \cdot |z_0|, 2n, 2^{n/4}\}$,
$M \leftarrow M \setminus \{z\colon |z| \leq m\}$ und gehe zum Schritt $n+1$.

Wie wir später sehen werden, ist es wichtig, vor Ausführung des Schrittes $n+1$ nach dem kleinsten Index $j_0 \in L'$ gesucht zu haben, für den es ein $z_0 \in \mathrm{Acc}(\mathrm{P}^A_{j_0}) \setminus B$ mit $|z_0| \leq 2^{(n-3)/4}$ gibt. Falls sich der Schrittindex $n_0$ wegen Durchlaufens von $\mathbf{M}_6$ vergrößert hat, müssen wir die Suche nach $j_0$ wiederholen, da sich bis zur größeren Schranke $2^{(n_0-3)/4}$ möglicherweise ein kleinerer Index $j_0$ finden läßt. Nur wenn das nicht der Fall ist, streichen wir $j_0$ aus unseren Listen und gehen zum Schritt $n+1$. Da wir alle Wörter bis zur Länge $m$ aus $M$ entfernen, kann $z_0$ später nicht mehr in $B$ gelangen (dies wäre nur bei gleichzeitiger Übernahme eines Wortes der Länge $2 \cdot |z_0|$ aus $M$ in $A$ möglich). Außerdem werden aus $M$ alle Wörter entfernt, die eventuell bei der Arbeit von $\mathrm{P}^{A_0}_{j_0}$ bzw. bei der Berechnung von $A_0$ als Orakelfragen auftauchten.

$\mathbf{M}_7$: Ist $n > \dfrac{m}{2}$ und $\sum\limits_{j \in L} p_j(n) < 2^{n-2}$?

*Nein:* Setze $M \leftarrow M \setminus \{0,1\}^{2n}$ und gehe zum nächsten Schritt $n+1$.

*Ja:* Setze $z \leftarrow$ kleinstes Wort $z$ (bezügl. der kanonischen Ordnung) mit $|z| = n$ und $1z \cdot \{0,1\}^{n-1} \subseteq M$
(wir werden anschließend beweisen, daß ein solches Wort stets existiert), $M \leftarrow M \setminus \bigcup\limits_{j \in L} Q(P^A_j, z)$, $i \leftarrow i+1$
und frage, ob es ein $j \in L$ gibt mit $z \in \mathrm{Acc}(\mathrm{P}^A_j)$.

*Ja:* ($z$ wird nicht in $B$ aufgenommen),
$L' \leftarrow L' \setminus \{j\colon z \in \mathrm{Acc}(\mathrm{P}^A_j)\}$,
$L \leftarrow (L \setminus \{j\colon z \in \mathrm{Acc}(\mathrm{P}^A_j)\}) \cup \{i\}$, $M \leftarrow M \setminus \{0,1\}^{2n}$
und gehe zum nächsten Schritt $n+1$.

*Nein:* $B \leftarrow B \cup \{z\}$, $A \leftarrow A \cup \{1zw\colon |w| = n-1 \wedge 1zw \in M\}$,
$L \leftarrow L \cup \{i\}$, $M \leftarrow M \setminus \{0,1\}^{2n}$
und gehe zum nächsten Schritt $n+1$.

Wir beweisen nun in den folgenden Punkten, daß die hier konstruierten Mengen $A$ und $B$ unseren Satz erfüllen.

1. Aus der Definition unserer Codierung folgt unmittelbar:

$$\text{Ist } |\text{cod}(i, x, l)| \leq 2^n, \text{ so ist } i < n,\ |x| < \frac{n}{4} \text{ und } l < 2^{n/4}.$$

2. $\mathbf{NEL}^A \subseteq \mathbf{EL}^A$. Ist $L \in \mathbf{NEL}^A$, so gibt es eine Maschine $\text{NEL}_i^A$ und eine lineare Funktion $f$, so daß $\text{NEL}_i^A$ in der Zeit $2^f$ läuft und $L$ akzeptiert. Dann gilt $x \in L \leftrightarrow \text{cod}(i, x, 2^{f(|x|)}) \in A$. Es ist $|\text{cod}(i, x, 2^{f(|x|)})| \leq 2^{g(|x|)}$ für eine geeignete lineare Funktion $g$ und damit $L \in \mathbf{EL}^A$.

3. Nach dem Satz von Hartmanis gibt es keine dünne Menge in $\mathbf{NP}^A \setminus \mathbf{P}^A$.

4. Die Frage $(*)$ in $\mathbf{M}_3$ wird unendlich oft mit *Ja* beantwortet. $L$ kann nur in $\mathbf{M}_7$ vergrößert werden, und $\mathbf{M}_7$ wird nur erreicht, falls $(*)$ mit *Ja* beantwortet wird. Damit bleibt $L$ beschränkt solange $(*)$ nicht mit *Ja* beantwortet wird. Dann ist aber

$$\sum_{\substack{j \in L \\ z \in B''}} p_j(|z|) \leq \sum_{\substack{j \in L \\ l \leq n/2}} 2^l \cdot p_j(l) < 2^{n/2+1} \cdot n^{\text{const}} < 2^{n-3}$$

für hinreichend großes $n$.

5. Jedes Element aus $B'$ gelangt irgendwann in $B$. Jedes Element aus $B'$ wird für hinreichend großes $n$ in $\mathbf{M}_3$ Element von $B''$. Wenn anschließend $(*)$ gemäß 4. mit *Ja* beantwortet wird, gelangen wir zu $\mathbf{M}_4$ und können $\mathbf{M}_4 - \mathbf{M}_6$ nur mit $B \leftarrow B \cup B''$ verlassen.

6. $B \in \mathbf{NP}^A$. Gemäß 5. und unserer Konstruktion ist $z \in B \leftrightarrow 1zw \in A$ für ein Wort $w$ mit $|w| = |z| - 1$. Das bedeutet $B \leq_{\mathrm{T}}^{\mathbf{NP}} A$ oder $B \in \mathbf{NP}^A$.

7. $\mathbf{M}_7$ wird unendlich oft erreicht und mit *Ja* beantwortet. Gemäß 4. erreichen wir unendlich oft $\mathbf{M}_4$ mit positiver Antwort auf $(*)$. Dann kann $\mathbf{M}_4 - \mathbf{M}_6$ nur verlassen werden zu $\mathbf{M}_7$ oder zu einem neuen Schritt bei gleichzeitiger Verringerung von $L$. $L$ ist aber endlich und kann nur in $\mathbf{M}_7$ vergrößert werden. $m$ kann ebenfalls nur bei gleichzeitiger Verringerung von $L$ vergrößert werden. Damit bleiben $L$ und $m$ beschränkt, solange $\mathbf{M}_7$ nicht positiv beantwortet wird, was aber dann für hinreichend großes $n$ stets der Fall ist.

8. Bei positiver Antwort auf $\mathbf{M}_7$ existiert stets ein Wort $z$ mit

$$|z| = n \text{ und } 1z \cdot \{0, 1\}^{n-1} \subseteq M.$$

Wir zählen die Wörter der Länge $2n$, die bis zum entsprechenden Teilschritt aus $M$ entfernt werden konnten. Wegen $n > m/2$ wurde kein Wort dieser Länge im Fall *c)* von $\mathbf{M}_5$ entfernt. In den Teilschritten vor $\mathbf{M}_3$ wurden bisher insgesamt höchstens so viele Wörter aus $M$ entfernt, wie während den ersten akzeptierende Berechnungen von $\text{NEL}_j^A$ auf $x$ mit maximal $l$ Schritten als Orakelfragen vorkommen konnten, wobei $|\text{cod}(j, x, l)| \leq 2^n$. Gemäß 1. und $n > 7$ sind das weniger als $n \cdot 2^{n/4} \cdot 2^{n/4} = n \cdot 2^{n/2} \leq 2^{n-1}$ Wörter. In allen bisheri-

gen. Teilschritten $\mathbf{M}_3$ wurden weniger als $\sum_{k=8}^{n} 2^{k-3} < 2^{n-2}$ Wörter der Länge $2n$ aus $M$ entfernt und in allen bisherigen Teilschritten $\mathbf{M}_7$ weniger als $\sum_{k=8}^{n-1} 2^{k-2} < 2^{n-2}$ derartige Wörter. Insgesamt wurden also weniger als $2^n$ Wörter der Länge $2n$ aus $M$ entfernt. Da es genau $2^n$ Mengen der Gestalt $1z \cdot \{0, 1\}^{n-1}$ mit $|z| = n$ gibt, muß wenigstens eine derartige Menge als Teilmenge von $M$ verbleiben. Nach Fixierung eines entsprechenden Wortes $z$ werden nochmals weniger als $2^{n-2}$ Wörter aus $M$ entfernt, und es muß wenigstens ein Wort $1zw$ mit $|w| = n-1$ in $M$ verbleiben.

9. Aus 7. folgt, daß jeder Index $i$ irgendwann ein Element von $L$ wird.

10. $B$ ist unendlich. Nehmen wir das Gegenteil an, so heißt das, daß ab einem gewissen Schritt $n_1$ die Frage in $\mathbf{M}_7$ nach einem Index $j \in L$ mit $z \in \mathrm{Acc}(\mathrm{P}_j^A)$ nur noch mit *Ja* beantwortet wird, wobei die entsprechenden Indizes aus $L$ entfernt werden. Damit bleibt die Anzahl von $L$ beschränkt. Andererseits gibt es unendlich viele Indizes $j$ mit $\mathrm{Acc}(\mathrm{P}_j^A) = \emptyset$, und diese Indizes werden nie aus $L$ entfernt. Zusammen mit 9. folgt die Unbeschränktheit der Anzahl von $L$. Damit muß $B$ unendlich sein.

11. Wird $j$ jemals aus $L$ entfernt, so ist $\mathrm{Acc}(\mathrm{P}_j^A) \not\subseteq B$. Dies folgt aus den Konstruktionsschritten $\mathbf{M}_7$ und Fall *c)* gemäß des dort gegebenen Kommentars.

12. Keine unendliche Teilmenge von $B$ ist in $\mathbf{P}^A$. Angenommen, es gibt $i$ mit $\mathrm{Acc}(\mathrm{P}_i^A) \subseteq B$ und $\mathrm{Acc}(\mathrm{P}_i^A)$ unendlich. Wir betrachten das kleinste derartige $i$. Gemäß 9. und 11. ist dieses $i$ ab einem gewissen Schritt $n_1$ stets in $L$. Dann kann nach dem Schritt $n_1$ kein $z \in \mathrm{Acc}(\mathrm{P}_i^A)$ in einem Teilschritt $\mathbf{M}_7$ zu $B$ kommen. D.h. $\mathrm{Acc}(\mathrm{P}_i^A)$ enthält nur solche Wörter mit Längen $\geq n_1$, die aus $B'$ über $B''$ in $B$ gelangen. Damit existiert $n_2 \geq n_1$ derart, daß in jedem Schritt ab $n_2$ $i \in L'$ gilt und für jedes $j < i$ ist entweder $j \notin L'$ oder $j$ bleibt für immer in $L'$ und $\mathrm{Acc}(P_j^A)$ ist endlich. Es sei nun $(B \cup B')_{n_2}$ derjenige Anteil der Menge $B \cup B'$, der bis zum Schritt $n_2$ erreicht wurde. Da $(B \cup B')_{n_2}$ endlich und $\mathrm{Acc}(\mathrm{P}_i^A)$ unendlich ist, existiert das bezüglich der kanonischen Ordnung kleinste Wort $z$ mit $z \in \mathrm{Acc}(\mathrm{P}_i^A) \setminus (B \cup B')_{n_2}$. Dieses $z$ komme in Schritt $s > n_2$ erstmals zu $B'$. Dann müssen entsprechende Wörter der Länge $2 \cdot |z|$ in $M_{\mathfrak{B}}^{+}$ sein, und diese erschienen als Orakelfragen innerhalb von $l$ Schritten der Arbeit von $\mathrm{NEL}_r^A$ auf $x$, wobei $|\mathrm{cod}(r, x, l)| \leq 2^s$. Gemäß 1. haben sie eine Länge kleiner als $l < 2^{s/4}$. Dann ist $|z| < 2^{s/4-1}$. Betrachten wir nun den Schritt $s-1$, so wurde entweder dieser Schritt vollständig ausgeführt oder im Fall *c)* von $\mathbf{M}_6$ übersprungen. Im ersten Falle gelangen wir mit $j_0 = i$ und $z_0 = z$ im Fall *c)* von $\mathbf{M}_6$ zum Schritt $s$, weil dort alle Wörter bis zur Länge $2^{(s-1-3)/4} = 2^{s/4-1}$ überprüft wurden. Im zweiten Falle mußte *c)* mit $n_0 = s-1 > n$ erreicht worden sein, und die Suche wurde bis zur Länge $2^{s/4-1}$ wiederholt. Aber dann gelangen wir ebenfalls mit $j_0 = i$ und $z_0 = z$ zum Schritt $s$. In beiden Fällen wurde $i$ aus $L$ entfernt, und gemäß 11. ist $\mathrm{Acc}(\mathrm{P}_i^A) \not\subseteq B$ im Widerspruch zu unserer Annahme.

Damit ist der Satz bewiesen. □

## LITERATUR

[1] *Baker T., Gill J.* and *Solovay R.:* Relativizations of the P = ? NP question. *SIAM J. Computing* **4** (1975), 431 – 442.
[2] *Balcázar J. L.:* Separating, strongly separating, and collapsing relativized complexity classes. In: Proc. MFCS 1984, Praha, Lecture Notes in Computer Science **176.** Springer-Verlag, Berlin – Heidelberg – New York – Tokyo, (1984), 1 – 16.
[3] *Book R. V., Wilson C. B.* and *Xu Mei-rui:* Relativizing time, space, and time-space. *SIAM J. Computing* **11** (1982), 571 – 581.
[4] *Cook S. A.:* The complexity of theorem-proving procedures. In: Proc. Third Annual ACM Symposium on Theory of Computing, 1971, 151 – 158.
[5] *Garey M. R.* and *Johnson D. S.:* Computers and Intractibility. A Guide to the Theory of NP-completeness. Freeman, San Francisco, 1979.
[6] *Hartmanis J.:* On sparse sets in NP – P. *Inform. Process. Lett.* **16** (2) (1983), 55 – 60.
[7] *Hopcroft J.* and *Ullman J.:* Formal Languages and Their Relation to Automata. Addison-Wesley, Reading (Massachusetts), 1969.
[8] *Kurtz S.:* On sparse sets in NP – P: Relativization. *SIAM J. Computing* **14** (1985), 113 – 119.
[9] *Lischke G.:* Oracle-constructions to prove all possible relationships between relativizations of P, NP, EL, NEL, EP and NEP. *Z. Math. Logik Grundl. Math.* **32** (1986), 257 – 270.
[10] *Schöning U.* and *Book R. V.:* Immunity. In: Proc. ICALP 1983, Barcelona. *Lecture Notes in Computer Science* **154.** Springer-Verlag, Berlin – Heidelberg – New York – Tokyo, 1983, 653 – 661.
[11] *Wagner K.* and *Wechsung G.:* Computational complexity. VEB Deutscher Verlag der Wissenschaften, Berlin, 1986.
[12] *Wilson C. B.:* Relativization, reducibilities, and the exponential hierarchy. M. S. thesis, University of Toronto, Toronto, 1980.