Information and Computation 209 (2011) 173–182

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/ic

Observations on complete sets between linear time and polynomial time

Armin Hemmerling

Ernst-Moritz-Arndt–Universität Greifswald, Institut für Mathematik und Informatik, Walther-Rathenau-Str. 47, D–17487 Greifswald, Germany

ARTICLE INFO

Article history:
Received 7 June 2010
Available online 7 December 2010

Keywords:
Many-one reducibility
Completeness
Determinism versus nondeterminism
Polynomial-time hierarchy
Linear-time hierarchy
Boolean hierarchy

ABSTRACT

There is a single set that is complete for a variety of nondeterministic time complexity classes with respect to related versions of m-reducibility. This observation immediately leads to transfer results for determinism versus nondeterminism solutions. Also, an upward transfer of collapses of certain oracle hierarchies, built analogously to the polynomial-time or the linear-time hierarchies, can be shown by means of uniformly constructed sets that are complete for related levels of all these hierarchies. A similar result holds for difference hierarchies over nondeterministic complexity classes. Finally, we give an oracle set relative to which the nondeterministic classes coincide with the deterministic ones, for several sets of time bounds, and we prove that the strictness of the tape-number hierarchy for deterministic linear-time Turing machines does not relativize.

0. Introduction and overview

The technique of many-one reducibility and related completeness results are dominated by polynomial-time and logarithmic-space reductions. These are commonly used for several complexity classes. In this paper, we consider classes defined by sets of time bounds satisfying certain regularity properties that can be fulfilled in defining a variety of complexity classes between linear and polynomial time. If one always considers the m-reducibility which is defined by the same set of bounds that determines the complexity class, then there is a language which is complete for all such nondeterministic classes. A first application of this crucial observation in Section 2 yields upward transfers of inclusions between certain complexity classes.

Our main interest is directed to oracle hierarchies defined by generalizing the constructions underlying the linear-time and the polynomial-time hierarchies. Section 3 prepares the fundamentals on relativization. A sequence of languages being complete for the related levels of all oracle hierarchies is specified in Section 4. It easily follows an upward transfer of collapses of the hierarchies. Quite similar results for the difference hierarchies over nondeterministic complexity classes are obtained in Section 5.

The final Section 6 deals with certain space complexity classes, where completeness remains defined by means of time bounds. We give a simple language that is complete for all such nondeterministic space classes. Taken as an oracle in time-bounded computations, it enforces the equality of nondeterministic and deterministic classes. Another application shows that the statement of strictness of the tape-number hierarchy for deterministic linear-time Turing machines does not relativize.

Even if all results follow by means of standard techniques of computational complexity theory, this paper provides a unifying view to questions and constructions around determinism versus nondeterminism and gives new relationships between them on different complexity levels.

E-mail address: hemmerli@uni-greifswald.de

0890-5401/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2010.11.027

1. Basic notions and facts

Throughout this paper, we restrict ourselves to *languages* $L \subseteq \mathbb{X}^*$ for the fixed two-letter alphabet $\mathbb{X} = \{0, 1\}$. This does not cause any loss of generality. For a set B of *complexity bounds* $\beta : \mathbb{N} \longrightarrow \mathbb{N}$, the related *(time) complexity class* is

$$\begin{aligned}&[\mathrm{N}]\mathrm{Time}(B) = \bigcup\nolimits_{\beta\in B}[\mathrm{N}]\mathrm{Time}(\beta), \quad \text{where}\\ &[\mathrm{N}]\mathrm{Time}(\beta) \ = \{L \subseteq \mathbb{X}^* :\ L \text{ is accepted by an [N]TM } \mathfrak{M} \text{ with } \mathrm{time}_{\mathfrak{M}}(w) \le \beta(|w|) \text{ for almost all } w \in \mathbb{X}^*\}.\end{aligned}$$

By an [N]TM, we understand a [nondeterministic] k-tape *Turing machine* with a distinguished read-only input tape and k read–write work tapes, on which symbols from an arbitrary working alphabet can be used. The acceptance of a language L and the time complexity measure $\mathrm{time}_{\mathfrak{M}} : \mathbb{X}^* \longrightarrow \mathbb{N}$ are defined as usual. For further details concerning basic notions, the reader is referred to textbooks like [4,15]. Here we only sketch some essentials.

By means of the brackets [. . .], two analogous sentences or statements are summarized by one formulation: one of these is obtained by deleting the brackets only, the other one by deleting the brackets and their contents, everywhere in the related context. In order to have this opportunity of abbreviation, the prefix D in the denotations of deterministic TMs and complexity classes is suppressed.

By $\mathrm{code}(\mathfrak{M}) \in \mathbb{X}^*$, we denote the *code* of an [N]TM $\mathfrak{M}$ which is defined in some standard way such that the components of $\mathfrak{M}$, as number of tapes, state set, working alphabet and transition function, can easily be extracted by another machine, e.g., in order to simulate $\mathfrak{M}$. One can suppose that the sets $\{\mathrm{code}(\mathfrak{M}) :\ \mathfrak{M} \text{ is an [N]TM}\}$ and $\{\mathrm{code}(\mathfrak{M}) :\ \mathfrak{M} \text{ is a } k\text{-tape [N]TM}\}$, for any $k \in \mathbb{N}$, are deterministically decidable in linear time, i.e., they belong to

$$\mathrm{Lin} = \mathrm{Time}(B_{\mathrm{lin}}), \quad \text{where} \quad B_{\mathrm{lin}} = \{\beta : \beta(n) = c \cdot n, \ \text{with some } c \in \mathbb{N}_+\}$$

denotes the set of *linear bounds*; let $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$.

Furthermore, it is supposed that the strings $\mathrm{code}(\mathfrak{M})$ start with a unique prefix or terminate with a unique suffix such that the single copies of $\mathrm{code}(\mathfrak{M})$ can easily be identified in concatenations of them like $\mathrm{code}(\mathfrak{M})^t$, for any $t \in \mathbb{N}_+$.

It is well-known that the number of tapes of NTMs accepting languages can be restricted to two, without enlarging the order of time complexity. More precisely:

Lemma 1. *If a language L is accepted by an arbitrary multi-tape NTM with a time bound β, then L can also be accepted by a 2-tape NTM in time $\mathcal{O}(\beta)$.*

To sketch an idea of proof, let $\mathfrak{M}$ be a k-tape NTM accepting L in time β. A simulating 2-tape NTM $\mathfrak{M}'$ first guesses, on one of its work tapes, a candidate of a sequence of states and tape symbols on the actually scanned cells as well as of signs indicating which of the possible reactions according to the transition relation has to be chosen in each step. Then, by means of the other work tape, $\mathfrak{M}'$ tries to verify that this sequence belongs to an accepting computation of $\mathfrak{M}$ on the given input. This is done by simulating the related behavior of $\mathfrak{M}$ on tape $\varkappa$, successively in special stages for each $\varkappa \in \{1, \dots, k\}$. All this is possible within the time bound $(k + 1) \cdot \beta(n)$. $\square$

Let $\langle \cdot, \cdot \rangle$ be a *pairing function* computable in linear time, with inverses (projections) computable in linear time, too.

To compute word functions (within certain complexity bounds), we employ deterministic TMs with an additional one-way write-only output tape on which the values of the functions have to be produced. A function $\beta : \mathbb{N} \longrightarrow \mathbb{N}$ is called *time-constructible* iff the word function $f_\beta : \mathbb{X}^* \longrightarrow \mathbb{X}^*$, defined by $f_\beta(w) = 1^{\beta(|w|)}$, is (deterministically) computable in time $\mathcal{O}(\beta)$. Equivalently, one could require that there is a deterministic multi-tape TM which halts on any input $w \in \mathbb{X}^*$ after $\mathcal{O}(\beta(|w|))$ steps such that the last work tape carries the inscription $1^{\beta(|w|)}$ then.

For functions $\beta_1, \beta_2 : \mathbb{N} \longrightarrow \mathbb{N}$ (or $\beta_1, \beta_2 : \mathbb{X}^* \longrightarrow \mathbb{N}$), let $\beta_1 \le_a \beta_2$ mean that $\beta_1(x) \le \beta_2(x)$ for almost all arguments x. Operations on numbers are naturally transferred to number functions, e.g., $(\beta_1 + \beta_2)(n) = \beta_1(n) + \beta_2(n)$ and $(c \cdot \beta)(n) = c \cdot \beta(n)$ for any constant $c \in \mathbb{N}$.

Definition 1. A non-empty set of bounds, B, is called *regular* iff the following hold:

(i) for all $\beta \in B$ and all $n \in \mathbb{N}$, $n \le \beta(n) \le \beta(n + 1)$;
(ii) for all $\beta \in B$, there is a time-constructible $\beta' \in B$ such that $\beta \le_a \beta'$;
(iii) for all $\beta, \beta' \in B$, there is a $\beta'' \in B$ such that $\beta + \beta' \circ \beta \le_a \beta''$.

From (i) and (iii) we immediately obtain:

(iv) for all $\beta, \beta' \in B$, there is a $\beta'' \in B$ such that $\beta + \beta' \le_a \beta''$;
(v) for all $\beta \in B$ and $c \in \mathbb{N}$, there is a $\beta' \in B$ such that $c \cdot \beta \le_a \beta'$;
(vi) for all $\beta, \beta' \in B$, there is a $\beta'' \in B$ such that $\beta' \circ \beta \le_a \beta''$.

Conversely, (iv) and (vi) imply (iii).

Examples of regular sets are B_{lin}, the set of linear bounds we defined above,

$$B_{\mathrm{pol}} = \{\beta : \beta(n) = n^k,\ \text{with some } k \in \mathbb{N}_+\},\quad \text{the set of } \textit{polynomial bounds}, \text{ and}$$
$$B_{\mathrm{qlin}} = \{\beta : \beta(n) = c \cdot n \cdot \log(n)^k,\ \text{with } c, k \in \mathbb{N}_+\},\quad \text{the set of } \textit{quasilinear bounds}.$$

Here let $\log(n)$ denotes the length of the binary expansion of n, i.e., $\log(0) = 0$, $\log(1) = 1$, $\log(2) = \log(3) = 2$, etc. The importance of quasilinear-time computations was first stressed in [11].

Conditions (i) and (ii) are rather natural and fulfilled by a variety of sets of bounds. The condition (iii) implies a certain robustness of the complexity classes, in particular, it follows their closedness under B-reducibility, cf. the next section. If B contains the exponential function $\beta_{\exp}(n) = 2^n$ or a function dominating this, then for any $\beta \in B$ there exists a time-constructible function dominating $2^{\beta(n)}$ in B, and it follows $\mathrm{NTime}(B) = \mathrm{Time}(B)$. Thus, our results are mainly interesting for sets of subexponential bounds.

An example of such a regular set of superpolynomial bounds is

$$B_{\mathrm{spol}} = \{\beta : \beta(n) = 2^{c \cdot \log(n)^k},\ \text{with some } c, k \in \mathbb{N}_+\}.$$

For bound sets B_1 and B_2, let $B_1 \leq_a B_2$ denote that to any $\beta_1 \in B_1$ there is a $\beta_2 \in B_2$ such that $\beta_1 \leq_a \beta_2$. So we have $B_{\mathrm{lin}} \leq_a B_{\mathrm{qlin}} \leq_a B_{\mathrm{pol}} \leq_a B_{\mathrm{spol}}$, and $B_{\mathrm{lin}} \leq_a B$ for any regular set of bounds B. Infinitely many sets of bounds between B_{lin} and B_{qlin} are given by

$$B_{\mathrm{q}^m\mathrm{lin}} = \{\beta : \beta(n) = c \cdot n \cdot (\log^{\langle m \rangle}(n))^k,\ \text{with } c, k \in \mathbb{N}_+\},$$

where $\log^{\langle m \rangle}$ denotes the m-fold iteration of log, for $m \in \mathbb{N}_+$. All these sets $B_{\mathrm{q}^m\mathrm{lin}}$ are regular, and it holds $B_{\mathrm{lin}} \leq_a \cdots \leq_a B_{\mathrm{q}^3\mathrm{lin}} \leq_a B_{\mathrm{q}^2\mathrm{lin}} \leq_a B_{\mathrm{q}^1\mathrm{lin}} = B_{\mathrm{qlin}}$.

2. A complete set and simple transfer results

For a regular bound set B, the related *B-reducibility* $\leq_B$ between languages is defined as the m-reducibility by means of word functions computable with time bounds from B. More precisely, for $L, L' \subseteq \mathbb{X}^*$,

$$L \leq_B L' \ \text{ iff there is a word function } f : \mathbb{X}^* \longrightarrow \mathbb{X}^* \text{ which is computable in time } \beta, \text{ for some } \beta \in B,$$
$$\text{such that for all } w \in \mathbb{X}^*:\ w \in L \text{ iff } f(w) \in L'.$$

Lemma 2. *The relation $\leq_B$ is reflexive and transitive. The complexity classes* $[\mathrm{N}]\mathrm{Time}(B)$ *are (downward) closed under $\leq_B$, i.e., if $L' \leq_B L$ and $L \in$* $[\mathrm{N}]\mathrm{Time}(B)$, *then $L' \in$* $[\mathrm{N}]\mathrm{Time}(B)$.

This follows immediately from properties (i) and (iii) of regular sets. For example, if a function reducing L' to L is computed in time β and L is accepted in time β', then L' can be accepted in time $\mathcal{O}(\beta + \beta' \circ \beta)$. □

A language $L \subseteq \mathbb{X}^*$ is said to be $\mathrm{NTime}(B)$-*complete* iff $L \in \mathrm{NTime}(B)$ and $L' \leq_B L$ for all $L' \in \mathrm{NTime}(B)$. It would be more precise to denote this property as B-completeness, since it is determined by the set of bounds but not by the complexity class. The introduced denotation, however, corresponds to the commonly used one, and its meaning should always be clear throughout this paper.

Complete sets enable us to express inclusions of nondeterministic classes in deterministic ones by the questions whether these special sets belong to the deterministic classes:

Proposition 1. *Let B_1 and B_2 be regular bound sets such that $B_1 \leq_a B_2$. If L is an* $\mathrm{NTime}(B_1)$*-complete language, then it holds:* $\mathrm{NTime}(B_1) \subseteq \mathrm{Time}(B_2)$ *iff* $L \in \mathrm{Time}(B_2)$.

The forward implication is trivial; the converse holds since $\mathrm{Time}(B_2)$ is closed under $\leq_{B_2}$, hence under $\leq_{B_1}$ too. □

If $B_1 = B_2 = B$ and L is an $\mathrm{NTime}(B)$-complete language, then by Proposition 1 it holds: $\mathrm{NTime}(B) = \mathrm{Time}(B)$ iff $L \in \mathrm{Time}(B)$. For example, we have $[\mathrm{N}]\mathrm{Time}(B_{\mathrm{pol}}) = [\mathrm{N}]\mathrm{P}$. These are the famous classes of *polynomial-time* computability, and $\leq_{B\mathrm{pol}}$ is the usual polynomial-time reducibility. For the *linear-time* classes $[\mathrm{N}]\mathrm{Lin} = [\mathrm{N}]\mathrm{Time}(B_{\mathrm{lin}})$ it is known that $\mathrm{NLin} \neq \mathrm{Lin}$, cf. [10]. Thus, no NLin-complete language belongs to Lin. For the *quasilinear-time* classes $[\mathrm{N}]\mathrm{Qlin} = [\mathrm{N}]\mathrm{Time}(B_{\mathrm{qlin}})$, the problem whether $\mathrm{NQlin} = \mathrm{Qlin}$ is open like the P versus NP problem, cf. [9].

Now it is a simple observation that there exists a single language which is $\mathrm{NTime}(B)$-complete for any regular set B. Let

$$V = \{\langle w, \mathrm{code}(\mathfrak{M})^t\rangle : w \in \mathbb{X}^*, |w| \leq t \in \mathbb{N}_+ \text{ and } \mathfrak{M} \text{ is a 2-tape NTM that accepts } w \text{ in } \leq t \text{ steps}\}.$$

Lemma 3. $V \in$ NLin.

Given an input $u \in \mathbb{X}^*$, let the NTM $\mathfrak{M}_V$ first check whether $u = \langle w, \text{code}(\mathfrak{M})^t \rangle$ for some $w \in \mathbb{X}^*$, a 2-tape NTM $\mathfrak{M}$ and $t \in \mathbb{N}_+$. Due to our conventions, this can be done deterministically in linear time. If u has the above form, let $\mathfrak{M}_V$ simulate at most t steps of $\mathfrak{M}$ on input w as follows:

> First, the input word w is copied to a work tape of $\mathfrak{M}_V$. Then $\mathfrak{M}_V$ uses two further work tapes to simulate the two work tapes of $\mathfrak{M}$, where the symbols of the working alphabet of $\mathfrak{M}$ are encoded by suitable binary strings of equal lengths $\leq |\text{code}(\mathfrak{M})|$, in order to avoid additional delays by shifting tape contents in the course of the simulation. The code of the actual state of $\mathfrak{M}$ can be noted on a fourth work tape of $\mathfrak{M}_V$.
> The crucial point of the simulation is that, for each step τ, a suitable action of $\mathfrak{M}$ is chosen by means of the τ-th copy of code($\mathfrak{M}$) within the component $\text{code}(\mathfrak{M})^t$ of the input u. So $\mathfrak{M}_V$ can perform the simulation of t steps of $\mathfrak{M}$ by a one-way sweeping over the component $\text{code}(\mathfrak{M})^t$ of input u, and the whole simulation does not need more than $\mathcal{O}(t \cdot |\text{code}(\mathfrak{M})|) = \mathcal{O}(|u|)$ steps.
> Let $\mathfrak{M}_V$ accept the input u as soon as $\mathfrak{M}$ has accepted w in the simulated run; if this has not occurred after the t steps have been simulated, let $\mathfrak{M}_V$ enter a cycle of work without halting. $\square$

Proposition 2. *V is* NTime(B)*-complete for any regular set of bounds B.*

By Lemma 3 and $B_{\text{lin}} \leq_a B$, we have $V \in \text{NTime}(B)$. If $L \in \text{NTime}(B)$, by Lemma 1 there is a 2-tape NTM $\mathfrak{M}$ accepting L with a time bound $\beta \in B$. By property (ii) of regularity, β can be assumed to be time-constructible. The word function $f_{\mathfrak{M}} : \mathbb{X}^* \longrightarrow \mathbb{X}^*$ defined by

$$f_{\mathfrak{M}}(w) = \langle w, \text{code}(\mathfrak{M})^{\beta(|w|)} \rangle$$

is computable in time $\mathcal{O}(\beta(|w|))$. Also, $w \in L$ iff $f_{\mathfrak{M}}(w) \in V$. Hence $L \leq_B V$. $\square$

An immediate consequence of Propositions 1 and 2 is an upward transfer of inclusions between certain complexity classes. Such transfer results are usually obtained by means of padding. This does not seem to apply to the following one, however.

Proposition 3. *Let B_1, B_2 and B_3 be regular sets of bounds such that $B_1 \leq_a B_2 \leq_a B_3$. Then*

$$\text{NTime}(B_1) \subseteq \text{Time}(B_2) \quad \textit{implies} \quad \text{NTime}(B_3) = \text{Time}(B_3).$$

If $\text{NTime}(B_1) \subseteq \text{Time}(B_2)$, then $V \in \text{Time}(B_2) \subseteq \text{Time}(B_3)$, but from $V \in \text{Time}(B_3)$ it follows $\text{NTime}(B_3) = \text{Time}(B_3)$. $\square$

Even if the supposition $B_1 \leq_a B_2$ has not been used in the proof, it was added in the statement in order to indicate the main direction of applications. The related remark applies to similar results in the sequel. The assertion of Proposition 3 may look more interesting in the converse direction: if, e.g., P $\neq$ NP, then NLin $\not\subseteq$ P. This is obtained by putting $B_1 = B_{\text{lin}}$ and $B_2 = B_3 = B_{\text{pol}}$. By [1], the languages from NLin can even be accepted by real-time NTMs. Thus, if P $\neq$ NP, then there is a nondeterministically real-time acceptable language which does not belong to P.

In a similar simple way, inclusions of co-classes in complexity classes can be transferred. For a class of languages, $\mathcal{L}$, let $\text{co}\mathcal{L} = \{\mathbb{X}^* \setminus L : L \in \mathcal{L}\}$. A language $L \subseteq \mathbb{X}^*$ is called coNTime(B)*-complete* iff it belongs to coNTime(B) and $L' \leq_B L$ for all $L' \in \text{coNTime}(B)$. Obviously, L is NTime(B)-complete iff its *complement*, $\overline{L} = \mathbb{X}^* \setminus L$, is coNTime($B$)-complete. Thus, by Proposition 2, $\overline{V}$ is coNTime(B)-complete for any regular bound set B.

Proposition 4. *Let B_1, B_2 and B_3 be regular sets of bounds such that $B_1 \leq_a B_2 \leq_a B_3$. Then*

$$\text{coNTime}(B_1) \subseteq \text{[N]Time}(B_2) \quad \textit{implies} \quad \text{coNTime}(B_3) = \text{[N]Time}(B_3).$$

Indeed, if $\text{coNTime}(B_1) \subseteq \text{[N]Time}(B_2)$, then $\overline{V} \in \text{[N]Time}(B_2) \subseteq \text{[N]Time}(B_3)$. Thus, $\text{coNTime}(B_3) \subseteq \text{[N]Time}(B_3)$, and it follows $\text{coNTime}(B_3) = \text{[N]Time}(B_3)$. $\square$

For example, if coNLin $\subseteq$ [N]P, then coNP $=$ [N]P.

3. Relativization and o-regularity

The results of the previous section relativize in a straightforward way. We sketch some features which are fundamental for the sequel. For a general discussion of relativization, see [6].

By an [N]OTM, we understand a [nondeterministic] *oracle Turing machine* $\mathfrak{M}$ in the usual sense. Then $\text{code}(\mathfrak{M}) \in \mathbb{X}^*$ is defined in a straightforward way. Note that code($\mathfrak{M}$), like $\mathfrak{M}$ too, does not depend on an oracle set. To perform a computation or to estimate the complexity, however, it is necessary to specify an oracle. If the work of $\mathfrak{M}$ with a special oracle set $A \subseteq \mathbb{X}^*$ is considered, we shall write $\mathfrak{M}^A$. The function $\text{time}_{\mathfrak{M}^A} : \mathbb{X}^* \longrightarrow \mathbb{N}$ describes the corresponding time complexity. Under the *relativized complexity classes*, we understand

$$[\mathrm{N}]\mathrm{Time}^A(B) = \bigcup_{\beta\in B}[\mathrm{N}]\mathrm{Time}^A(\beta), \quad \text{where}$$
$$[\mathrm{N}]\mathrm{Time}^A(\beta) = \{L \subseteq \mathbb{X}^* : L \text{ is accepted by an [N]OTM } \mathfrak{M} \text{ with oracle set } A \text{ such that } \mathrm{time}_{\mathfrak{M}^A}(w) \le \beta(|w|) \text{ for almost all } w \in \mathbb{X}^*\}.$$

The classes $[\mathrm{N}]\mathrm{Lin}^A$, $[\mathrm{N}]\mathrm{Qlin}^A$ and $[\mathrm{N}]\mathrm{P}^A$ are straightforwardly defined by means of the bound sets B_{lin}, B_{qlin} and B_{pol}, respectively.

Lemma 1 relativizes, hence in accepting languages from $\mathrm{NTime}^A(B)$ we can restrict ourselves to 2-tape NOTMs, i.e., NOTMs with an input tape, a special oracle tape and two work tapes, time-bounded by functions from B.

Also in investigating relativized complexity classes, the un-relativized reducibilities $\le_B$ will be employed, however. Hence, a language $L \subseteq \mathbb{X}^*$ is briefly called $\mathrm{NTime}^A(B)$-*complete* iff $L \in \mathrm{NTime}^A(B)$ and $L' \le_B L$ for all $L' \in \mathrm{NTime}^A(B)$. Since the classes $[\mathrm{N}]\mathrm{Time}^A(B)$ are closed under $\le_B$, Proposition 1 relativizes:

Proposition 5. *For regular bound sets $B_1 \le_a B_2$ and an $\mathrm{NTime}^A(B_1)$-complete language L, it holds:*

$$\mathrm{NTime}^A(B_1) \subseteq \mathrm{Time}^A(B_2) \quad \textit{iff} \quad L \in \mathrm{Time}^A(B_2). \quad \square$$

For any $A \subseteq \mathbb{X}^*$, let

$$V^A = \{\langle w, \mathrm{code}(\mathfrak{M})^t\rangle : w \in \mathbb{X}^*, |w| \le t \in \mathbb{N}_+ \text{ and } \mathfrak{M} \text{ is a 2-tape NOTM such that } \mathfrak{M}^A \text{ accepts } w \text{ in} \le t \text{ steps}\}.$$

Proposition 6. *For any $A \subseteq \mathbb{X}^*$ it holds $V^A \in \mathrm{NLin}^A$, and V^A is $\mathrm{NTime}^A(B)$-complete if B is a regular set of bounds.*

This follows by relativizing the proofs of Lemma 3 and Proposition 2. Obviously, the reducing function $f_{\mathfrak{M}}$, where $f_{\mathfrak{M}}(w) = \langle w, \mathrm{code}(\mathfrak{M})^{\beta(|w|)}\rangle$ for a 2-tape NOTM $\mathfrak{M}$ and a time-constructible bound β, can be computed oracle-free in time $\mathcal{O}(\beta(|w|))$. $\square$

It immediately follows

Corollary 1. *Let $A \subseteq \mathbb{X}^*$ and $B_1 \le_a B_2 \le_a B_3$ for regular bound sets B_1, B_2, B_3. Then*

$$\mathrm{NTime}^A(B_1) \subseteq \mathrm{Time}^A(B_2) \quad \textit{implies} \quad \mathrm{NTime}^A(B_3) = \mathrm{Time}^A(B_3), \quad \textit{and}$$
$$\mathrm{coNTime}^A(B_1) \subseteq [\mathrm{N}]\mathrm{Time}^A(B_2) \quad \textit{implies} \quad \mathrm{coNTime}^A(B_3) = [\mathrm{N}]\mathrm{Time}^A(B_3). \quad \square$$

For example, from $\mathrm{P}^A \ne \mathrm{NP}^A$ it follows $\mathrm{NLin}^A \not\subseteq \mathrm{P}^A$. On the other hand, if $\mathrm{Lin}^A = \mathrm{NLin}^A$, then $\mathrm{P}^A = \mathrm{NP}^A$. An oracle set A with $\mathrm{Lin}^A = \mathrm{NLin}^A$ will be given in Section 6.

Now it is necessary to determine the way of working of [N]OTMs on their oracle tapes more precisely. As usual, we assume that the oracle tape is always erased after an oracle query has been asked. Hence, to prepare the next oracle step, the machine has to generate the new query completely, starting from the empty oracle tape. So the sum of lengths of all oracle queries within a computation of t steps is bounded by the number t. Another regime of working, where an oracle query does not change the previous content of the oracle tape, is discussed in [5]. Propositions 5 and 6 and Corollary 1 hold also in this setting.

According to our understanding, we have $[\mathrm{N}]\mathrm{Lin}^A \subseteq [\mathrm{N}]\mathrm{Lin}$ if $A \in \mathrm{Lin}$. Indeed, all queries to oracle A can be replaced by the decision procedure of A. So, from an [N]OTM $\widehat{\mathfrak{M}}$ with $\mathrm{time}_{\widehat{\mathfrak{M}}^A}(w) \le_a \widehat{\beta}(|w|)$, one obtains an [N]TM $\widetilde{\mathfrak{M}}$ accepting the same language as $\widehat{\mathfrak{M}}$ such that, for almost all $w \in \mathbb{X}^*$,

$$\mathrm{time}_{\widetilde{\mathfrak{M}}}(w) \le \widehat{\beta}(|w|) + \sum_{i=1}^{l} \beta(|y_i|),$$

if $y_1, \ldots, y_l$ are the oracle queries asked by $\widehat{\mathfrak{M}}$ in the course of a shortest accepting computation on input w, and β is a time bound of a (deterministic) TM deciding A. Since $\sum_{i=1}^{l} |y_i| \le \widehat{\beta}(|w|)$, from $\beta, \widehat{\beta} \in B_{\mathrm{lin}}$ it follows that $\mathrm{time}_{\widetilde{\mathfrak{M}}}(w) \le_a \beta'(|w|)$, for some $\beta' \in B_{\mathrm{lin}}$:
if $\widehat{\beta}(n) = \widehat{c} \cdot n$, $\beta(n) = c \cdot n$, $n = |w|$ and $m_i = |y_i|$, we have

$$\begin{aligned}\widehat{\beta}(n) + \sum_{i=1}^{l} \beta(m_i) &= \widehat{c} \cdot n + \sum_{i=1}^{l} c \cdot m_i \\ &\le \widehat{c} \cdot n + c \cdot \widehat{c} \cdot n = \widehat{c} \cdot (1 + c) \cdot n.\end{aligned}$$

In order to enable us to use this technique of replacing oracle queries by decisions, we put a further condition which leads to the notion of o-regular bound sets. The denotation can be understood as an abbreviation of "oracle-regular".

Definition 2. A regular set B of complexity bounds is said to be *o-regular* iff it fulfills:

(vii) for all $\beta \in B$, there is a $\beta' \in B$ such that whenever $\sum_{i=1}^{l} m_i \le \beta(n)$ for some $l, m_1, \ldots, m_l, n \in \mathbb{N}_+$, then $\sum_{i=1}^{l} \beta(m_i) \le \beta'(n)$.

Applying this property to the above idea of replacing oracle queries by decisions, we get: $\mathrm{time}_{\widetilde{\mathfrak{M}}}(w) \le_a \widehat{\beta}(|w|) + \beta'(|w|) \le_a \beta''(|w|)$, for some $\beta'' \in B$, the latter because of condition (iv) of regularity.

For a class $\mathcal{A}$ of oracle sets and a bound set B, let

$$[\mathrm{N}]\mathrm{Time}^{\mathcal{A}}(B) = \bigcup_{A \in \mathcal{A}} [\mathrm{N}]\mathrm{Time}^{A}(B).$$

So we have shown

Lemma 4. *If B is an o-regular set of bounds, then* $[\mathrm{N}]\mathrm{Time}^{\mathrm{Time}(B)}(B) = [\mathrm{N}]\mathrm{Time}(B)$. $\square$

One easily verifies that the bound sets we dealt with so far, B_{pol}, B_{lin}, B_{qlin}, B_{spol} and $B_{\mathrm{q}^m\mathrm{lin}}$, are o-regular. This also enables us to replace a class $\mathcal{A}$ of oracles by an $\mathcal{A}$-complete set A:

Proposition 7. *Let B be an o-regular set of bounds, $A \in \mathcal{A}$, and $A' \le_B A$ for all $A' \in \mathcal{A}$. Then it holds:*

$$[\mathrm{N}]\mathrm{Time}^{\mathcal{A}}(B) = [\mathrm{N}]\mathrm{Time}^{A}(B).$$

To show this, assume that $L \in [\mathrm{N}]\mathrm{Time}^{A'}(B)$, for some $A' \in \mathcal{A}$, and $\mathfrak{M}$ is an [N]OTM such that $\mathfrak{M}^{A'}$ accepts L with a time bound $\beta' \in B$. Furthermore, let $A' \le_B A$ via some word function f which can be computed with a time bound $\beta \in B$. Then the oracle queries of the form "$y_i \in A'$?" in $\mathfrak{M}^{A'}$-computations can be replaced by a procedure computing $f(y_i)$ and asking "$f(y_i) \in A$?". In this way we get an [N]OTM $\widetilde{\mathfrak{M}}$ such that $\widetilde{\mathfrak{M}}^{A}$ accepts L too, and

$$\mathrm{time}_{\widetilde{\mathfrak{M}}^A}(w) \le \beta'(|w|) + \sum_{i=1}^{l} \beta(|y_i|), \quad \text{for almost all input words } w,$$

with oracle queries $y_1, \ldots, y_l$ such that $\sum_{i=1}^{l} |y_i| \le \beta'(|w|)$. By condition (iv) of regularity, there is a bound $\beta'' \in B$ with $\beta + \beta' \le_a \beta''$. Then $\sum_{i=1}^{l} |y_i| \le \beta''(|w|)$ for almost all inputs w, and by condition (vii) there is a $\beta''' \in B$ such that $\sum_{i=1}^{l} \beta(|y_i|) \le \sum_{i=1}^{l} \beta''(|y_i|) \le \beta'''(|w|)$. Thus,

$$\mathrm{time}_{\widetilde{\mathfrak{M}}^A}(w) \le_a \beta'(|w|) + \beta'''(|w|) \le_a \widetilde{\beta}(|w|),$$

with a suitable bound $\widetilde{\beta} \in B$ which exists due to condition (iv) applied to β' and β'''. $\square$

4. Complete sets in oracle hierarchies

By *oracle hierarchies*, we understand hierarchies defined inductively by means of time-bounded NOTMs analogously to the polynomial-time hierarchy [12,13] or the linear-time hierarchy [16]. More precisely, for any set of bounds, B, let

$$\Sigma_1^B = \mathrm{NTime}(B),$$

$$\Sigma_{k+1}^B = \mathrm{NTime}^{\Sigma_k^B}(B), \quad \text{for } k \ge 1, \text{ and}$$

$$\Pi_k^B = \mathrm{co}\Sigma_k^B = \{\mathbb{X}^* \setminus A : A \in \Sigma_k^B\}, \quad \text{for } k \in \mathbb{N}_+ .$$

Obviously, $\Sigma_k^B \cup \Pi_k^B \subseteq \Sigma_{k+1}^B \cap \Pi_{k+1}^B$, for all $k \in \mathbb{N}_+$, if B contains a function β such that $\beta(n) \ge n$. We put

$$\mathrm{OH}^B = \bigcup_{k \in \mathbb{N}_+} \Sigma_k^B.$$

For $B = B_{\mathrm{pol}}$, the usual polynomial-time hierarchy is obtained in this way: $\mathrm{PH} = \bigcup_{k \in \mathbb{N}_+} \Sigma_k^{\mathrm{pol}}$, where $\Sigma_k^{\mathrm{pol}} = \Sigma_k^{B_{\mathrm{pol}}}$; $B = B_{\mathrm{lin}}$ yields the linear-time hierarchy, $\mathrm{LinH} = \bigcup_{k \in \mathbb{N}_+} \Sigma_k^{\mathrm{lin}}$, where $\Sigma_k^{\mathrm{lin}} = \Sigma_k^{B_{\mathrm{lin}}}$; and $B = B_{\mathrm{qlin}}$ yields the quasilinear-time hierarchy, $\mathrm{QlinH} = \bigcup_{k \in \mathbb{N}_+} \Sigma_k^{\mathrm{qlin}}$, where $\Sigma_k^{\mathrm{qlin}} = \Sigma_k^{B_{\mathrm{qlin}}}$. The latter one was studied in [9]. It is open, whether these hierarchies "are infinite" or collapse to one of their levels Σ_k^B.

The *join* of languages $A_0, A_1 \subseteq \mathbb{X}^*$ is defined by

$$A_0 \oplus A_1 = \{0w : w \in A_0\} \cup \{1w : w \in A_1\}.$$

Lemma 5. *For any o-regular bound set B and $k \in \mathbb{N}_+$, the classes Σ_k^B and Π_k^B are closed under $\le_B$ as well as under union, intersection and join of any two sets. Moreover, it holds:* $\mathrm{OH}^B = \Sigma_k^B$ *iff* $\Sigma_k^B = \Pi_k^B$ *iff* $\Sigma_k^B \subseteq \Pi_k^B$ *iff* $\Sigma_k^B \supseteq \Pi_k^B$.

The closure properties even hold for all complexity classes [N]Time$^A(B)$ if B is a regular set of bounds. Also, it is obvious that $\Sigma_k^B = \Pi_k^B$ is equivalent to $\Sigma_k^B \subseteq \Pi_k^B$ and to the converse inclusion, as well as that $\mathrm{OH}^B = \Sigma_k^B$ implies $\Sigma_k^B = \Pi_k^B$. Now suppose $\Sigma_k^B = \Pi_k^B$. We shall show that $\Sigma_{k+1}^B = \Sigma_k^B$. This implies $\Pi_{k+1}^B = \Pi_k^B$, hence $\Sigma_{k+l}^B = \Pi_{k+l}^B = \Sigma_k^B$ for all $l \in \mathbb{N}$, what completes the proof of Lemma 5.

If $L \in \Sigma_{k+1}^B$, it is accepted by $\widehat{\mathfrak{M}}^A$ with a time bound $\widehat{\beta} \in B$, for an NOTM $\widehat{\mathfrak{M}}$ and an oracle set $A \in \Sigma_k^B$. By supposition, $\overline{A} = \mathbb{X}^* \setminus A \in \Sigma_k^B$, and there are NOTMs $\mathfrak{M}$ and $\overline{\mathfrak{M}}$ such that $\mathfrak{M}^{A_0}$ accepts A and $\overline{\mathfrak{M}}^{A_1}$ accepts $\overline{A}$, both with time bounds β_0 and β_1, respectively, from B and oracles $A_0, A_1 \in \Sigma_{k-1}^B$. (For the rest of the proof, we assume that $k \geq 2$. The case $k = 1$ can analogously be treated.) By the first part of the lemma, we have $A_0 \oplus A_1 \in \Sigma_{k-1}^B$, too. Now let the NOTM $\widetilde{\mathfrak{M}}$, with the oracle set $A_0 \oplus A_1$, simulate $\widehat{\mathfrak{M}}^A$ in such a way that the oracle queries are replaced by a subroutine that guesses answers and verifies these by simulating $\mathfrak{M}^{A_0}$ and $\overline{\mathfrak{M}}^{A_1}$, respectively. To almost every input $w \in L$, there is an accepting $\widehat{\mathfrak{M}}^A$-computation that only puts oracle queries $y_1, \ldots, y_l$ such that $\sum_{i=1}^l |y_i| \leq \widehat{\beta}(|w|)$. Since B is regular, there is a $\beta \in B$ such that $\beta_0 + \beta_1 + \widehat{\beta} \leq_a \beta$. Hence, for almost all $w \in L$, $\sum_{i=1}^l |y_i| \leq \beta(|w|)$, and by condition (vii) there is a $\beta' \in B$ such that $\sum_{i=1}^l \beta(|y_i|) \leq \beta'(|w|)$. Thus, for almost all inputs $w \in L$, there are $\widetilde{\mathfrak{M}}^{A_0 \oplus A_1}$-computations accepting w in $\mathcal{O}(\beta')$ steps, and we have $L \in \mathrm{NTime}^{A_0 \oplus A_1}(B) \subseteq \Sigma_k^B$. $\square$

The completeness of sets in the classes of oracle hierarchies is again understood with respect to the B-reducibilities, i.e., a language L is called Σ_k^B-*complete* iff $L \in \Sigma_k^B$ and $L' \leq_B L$ for all $L' \in \Sigma_k^B$. Σ_k^B-complete languages allow us to represent the classes of the oracle hierarchy as complexity classes relativized to a special oracle and to express the collapse of OH^B to Σ_k^B by the question whether such a special set belongs to Π_k^B:

Lemma 6. *Let B be an o-regular set of bounds, $k \in \mathbb{N}_+$ and L be a Σ_k^B-complete language. Then $\Sigma_{k+1}^B = \mathrm{NTime}^L(B)$, and it holds: $\mathrm{OH}^B = \Sigma_k^B$ iff $L \in \Pi_k^B$.*

The first assertion follows by Proposition 7, the second one can easily be proved by means of Lemma 5. $\square$

Surprisingly, there are sets V_k that are Σ_k^B-complete for a great variety of bound sets B. To show this, we employ the universal NTime(B)-complete set V from Section 2 and its relativized modifications V^A introduced in Section 3. Let

$$V_1 = V, \quad \text{and}$$
$$V_{k+1} = V^{V_k} \quad \text{for all } k \in \mathbb{N}_+ .$$

Proposition 8. *For any o-regular set of bounds, B, and all $k \in \mathbb{N}_+$ it holds: V_k is Σ_k^B-complete.*

The proof is by induction on k. For $k = 1$, the assertion holds by Proposition 2. Suppose that V_k is Σ_k^B-complete. By Proposition 7, we have $\Sigma_{k+1}^B = \mathrm{NTime}^{V_k}(B)$. Now Proposition 6 yields that $V_{k+1} = V^{V_k}$ is $\mathrm{NTime}^{V_k}(B)$-complete, i.e., Σ_{k+1}^B-complete. $\square$

We are going to apply this result to oracle hierarchies. First, a simple monotony property should be noticed. It easily follows by induction:

Lemma 7. *If $B_1 \leq_a B_2$ for bound sets B_1 and B_2, then $\mathrm{OH}^{B_1} \subseteq \mathrm{OH}^{B_2}$, and, moreover, $\Sigma_k^{B_1} \subseteq \Sigma_k^{B_2}$ and $\Pi_k^{B_1} \subseteq \Pi_k^{B_2}$ for all $k \in \mathbb{N}_+$.* $\square$

Now we are able to show a transfer result for collapses of oracle hierarchies. This could also be proved by means of padding, under a further natural assumption on the bound sets. Within our framework, however, it follows immediately.

Proposition 9. *Let $B_1 \leq_a B_2$ for o-regular bound sets B_1 and B_2. If $\mathrm{OH}^{B_1} = \Sigma_k^{B_1}$ for some $k \in \mathbb{N}_+$, then $\mathrm{OH}^{B_2} = \Sigma_k^{B_2}$.*

Indeed, by Lemma 6 and Proposition 8, $\mathrm{OH}^{B_1} = \Sigma_k^{B_1}$ implies $V_k \in \Pi_k^{B_1}$. Since $B_1 \leq_a B_2$, by Lemma 7 it follows $V_k \in \Pi_k^{B_2}$, and this yields $\mathrm{OH}^{B_2} = \Sigma_k^{B_2}$. $\square$

For example, a collapse of the linear-time hierarchy or of the quasilinear-time hierarchy would imply a collapse of the polynomial-time hierarchy. It might be still more impressive to realize that already if a lower hierarchy is completely contained in some level of a higher oracle hierarchy, the latter collapses:

Proposition 10. *Let $B_1 \leq_a B_2$ for o-regular bound sets B_1 and B_2. If, for some $k \in \mathbb{N}_+$, $\Sigma_{k+1}^{B_1} \subseteq \Sigma_k^{B_2} \cup \Pi_k^{B_2}$, then $\mathrm{OH}^{B_2} = \Sigma_{k+1}^{B_2}$.*

Indeed, $\Sigma_{k+1}^{B_1} \subseteq \Sigma_k^{B_2} \cup \Pi_k^{B_2}$ yields $V_{k+1} \in \Sigma_k^{B_2} \cup \Pi_k^{B_2} \subseteq \Pi_{k+1}^{B_2}$, hence $\mathrm{OH}^{B_2} = \Sigma_{k+1}^{B_2}$ by Lemma 6. $\square$

For example, if the polynomial-time hierarchy does not collapse, then the linear-time hierarchy cannot be contained in any Σ_k^{pol}, and even $\Sigma_2^{\mathrm{lin}} \subseteq \Sigma_1^{\mathrm{pol}} \cup \Pi_1^{\mathrm{pol}} = \mathrm{NP} \cup \mathrm{coNP}$ is impossible.

5. Transfer of collapses of difference hierarchies

This section is another continuation of Section 2. The results follow even more directly than those of Section 4. In particular, no relativization is needed. We preferred to deal with the oracle hierarchies first, since they are older and surely of a broader interest than the difference hierarchies which are the subject of this section. On the other hand, in some respect, now the treatment is analogous to that of the previous section such that we can omit the elaborations of several details.

For a regular set of bounds, B, let the corresponding *difference hierarchy* consist of the classes D_k^B, for $k \in \mathbb{N}_+$, which are defined by

$$D_k^B = \begin{cases} \{(A_1 \setminus A_2) \cup \cdots \cup (A_{k-2} \setminus A_{k-1}) \cup A_k : A_1, \ldots, A_k \in \mathrm{NTime}(B)\} & \text{if } k \text{ is odd,} \\ \{(A_1 \setminus A_2) \cup \cdots \cup (A_{k-1} \setminus A_k) : \quad A_1, \ldots, A_k \in \mathrm{NTime}(B)\} & \text{if } k \text{ is even.} \end{cases}$$

Without loss of generality, one could additionally require that $A_1 \supseteq A_2 \supseteq \cdots \supseteq A_{k-1} \supseteq A_k$. $\mathrm{DH}^B = \bigcup_{k \in \mathbb{N}_+} D_k^B$ is exactly the Boolean closure of $\mathrm{NTime}(B)$ which, due to the regularity of B, is closed under union and intersection. So DH^B or the sequence of the classes D_k^B are also known as *Boolean hierarchy* (over $\mathrm{NTime}(B)$), in particular for the case $B = B_{\mathrm{pol}}$, where it was first introduced, cf. [2] for further details and references.

In analogy to the construction of V, there are natural languages expected to be complete in DH^B. Let $\langle \cdot, \ldots, \cdot \rangle$ denote the encoding of finite sequences of words by single words which is straightforwardly defined by means of the pairing function. Now we consider the sets

$$\begin{aligned} VD_k = \{\langle w, \mathrm{code}(\mathfrak{M}_1)^t, \ldots, \mathrm{code}(\mathfrak{M}_k)^t \rangle :\ & w \in \mathbb{X}^*, |w| \le t \in \mathbb{N}_+ \text{ and } \mathfrak{M}_1, \ldots, \mathfrak{M}_k \text{ are 2-tape NTMs} \\ & \text{such that } \mathfrak{M}_{2i-1} \text{ accepts } w \text{ in } \le t \text{ steps but } \mathfrak{M}_{2i} \text{ does not accept } w \text{ in } \le t \text{ steps,} \\ & \text{for some } i \in \mathbb{N}_+ \text{ with } 2i \le k, \text{ or } k \text{ is odd and } \mathfrak{M}_k \text{ accepts } w \text{ in } \le t \text{ steps}\}. \end{aligned}$$

If one puts

$$\begin{aligned} A_i^V = \{\langle w, \mathrm{code}(\mathfrak{M}_1)^t, \ldots, \mathrm{code}(\mathfrak{M}_k)^t \rangle :\ & w \in \mathbb{X}^*, |w| \le t \in \mathbb{N}_+ \text{ and} \\ & \mathfrak{M}_1, \ldots, \mathfrak{M}_k \text{ are 2-tape NTMs such that } \mathfrak{M}_i \text{ accepts } w \text{ in } \le t \text{ steps}\}, \end{aligned}$$

then it immediately follows

$$VD_k = \begin{cases} (A_1^V \setminus A_2^V) \cup \cdots \cup (A_{k-2}^V \setminus A_{k-1}^V) \cup A_k^V & \text{if } k \text{ is odd,} \\ (A_1^V \setminus A_2^V) \cup \cdots \cup (A_{k-1}^V \setminus A_k^V) & \text{if } k \text{ is even.} \end{cases}$$

Analogously to Lemma 3, one can show that $A_i^V \in \mathrm{NLin} \subseteq \mathrm{NTime}(B)$. Hence we have $VD_k \in D_k^B$ for any regular bound set B and $k \in \mathbb{N}_+$. On the other hand, let $L \in D_k^B$, i.e.,

$$L = \begin{cases} (A_1 \setminus A_2) \cup \cdots \cup (A_{k-2} \setminus A_{k-1}) \cup A_k & \text{if } k \text{ is odd,} \\ (A_1 \setminus A_2) \cup \cdots \cup (A_{k-1} \setminus A_k) & \text{if } k \text{ is even,} \end{cases}$$

for arbitrary languages $A_i \in \mathrm{NTime}(B)$ which are accepted by 2-tape NTMs $\mathfrak{M}_i$ with time bounds $\beta_i \in B$. Due to conditions (ii) and (iv) of regularity, there is a time-constructible $\beta \in B$ such that $\beta_i \le_a \beta$ for $1 \le i \le k$. Then the word function $f : \mathbb{X}^* \longrightarrow \mathbb{X}^*$ defined by

$$f(w) = \langle w, \mathrm{code}(\mathfrak{M}_1)^{\beta(|w|)}, \ldots, \mathrm{code}(\mathfrak{M}_k)^{\beta(|w|)} \rangle$$

is computable in time $\mathcal{O}(\beta(|w|))$, and it is an m-reduction of L to VD_k. So we have shown

Proposition 11. *For all regular sets of bounds, B, and $k \in \mathbb{N}_+$, the sets VD_k are D_k^B-complete with respect to $\le_B$.* □

Since these D_k^B-complete sets do not depend on the regular bound sets B, we get an upward transfer of collapses of the difference hierarchies. Of course, one has to employ the following facts which are easily shown.

Lemma 8. *For any regular bound set B, the classes D_k^B are closed under $\le_B$. If the language L is D_{k+1}^B-complete with respect to $\le_B$, then it holds: $\mathrm{DH}^B = D_k^B$ iff $D_{k+1}^B = D_k^B$ iff $L \in D_k^B$.* □

From this, we immediately obtain an upward transfer of collapses analogously to Propositions 9 and 10.

Proposition 12. *Let $B_1 \le_a B_2$ for regular sets of bounds, B_1 and B_2. If $\mathrm{DH}^{B_1} = D_k^{B_1}$ or only $D_{k+1}^{B_1} \subseteq D_k^{B_2}$ for some $k \in \mathbb{N}_+$, then $\mathrm{DH}^{B_2} = D_k^{B_2}$.* □

By [8] it is known that a collapse of the difference hierarchy over NP implies the collapse of the polynomial-time hierarchy to its third level. This yields

Corollary 2. *If the linear-time difference hierarchy collapses, i.e.,* $\mathrm{DH}^{B_{\mathrm{lin}}} = D_k^{B_{\mathrm{lin}}}$ *for some* $k \in \mathbb{N}_+$*, or if* $D_{k+1}^{B_{\mathrm{lin}}} \subseteq D_k^{B_{\mathrm{pol}}}$ *for some* $k \in \mathbb{N}_+$*, then* $\mathrm{PH} \subseteq \Sigma_3^{\mathrm{pol}}$. □

6. Time completeness in space classes

We omit a detailed treatment of space classes analogously to what we did for time complexity classes. In particular, oracle hierarchies over the interesting space classes collapse to their first level, since the nondeterministic space classes are closed under complement, see [7,14]. Here we only want to point out that there is a special set VS which is complete in several nondeterministic space classes under the related time-bounded reducibilities.

For a set of bounds, B, we consider the *space complexity class*

$$[\mathrm{N}]\mathrm{Space}(B) = \bigcup\nolimits_{\beta\in B}[\mathrm{N}]\mathrm{Space}(\beta), \quad \text{where}$$

$$[\mathrm{N}]\mathrm{Space}(\beta) = \{L \subseteq \mathbb{X}^* : L \text{ is accepted by an [N]TM } \mathfrak{M} \text{ with } \mathrm{space}_{\mathfrak{M}}(w) \le \beta(|w|) \text{ for almost all } w \in \mathbb{X}^*\}.$$

If $B_{\mathrm{lin}} \le_a B$, it suffices to consider *simple* [N]TMs in accepting languages from [N]Space(B), i.e., TMs with only one tape on which the input is given and the computation has to be performed, with a space bound $\beta \in B$.

For example, [N]LBA = [N]Space(B_{lin}) is the class of languages accepted by so-called [nondeterministic] *linear bounded automata*, and NLBA coincides with the class of context-sensitive languages over $\mathbb{X}$. NPSPACE = PSPACE = [N]Space(B_{pol}) consists of the languages acceptable in polynomial space.

Our universal complete set is defined analogously to V in Section 2. Let

$$VS = \{\langle w, \mathrm{code}(\mathfrak{M})^s\rangle : w \in \mathbb{X}^*, |w| \le s \in \mathbb{N}_+ \text{ and } \mathfrak{M} \text{ is a simple NTM that accepts } w \text{ using} \le s \text{ tape cells}\}.$$

Lemma 9. $VS \in$ NLBA.

Given an input $u \in \mathbb{X}^*$, a linear space-bounded TM first checks that $u = \langle w, \mathrm{code}(\mathfrak{M})^s\rangle$ for some $w \in \mathbb{X}^*$, $|w| \le s \in \mathbb{N}_+$, and a simple NTM $\mathfrak{M}$. Then it simulates the work of $\mathfrak{M}$ on input w up to an accepting configuration, but only as long as the related $\mathfrak{M}$-computation on w uses at most s cells. □

Proposition 13. *For any regular bound set B, VS is* NSpace(B)*-complete with respect to* $\le_B$.

By Lemma 9, $VS \in$ NSpace(B) . For $L \in$ NSpace(B), let $\mathfrak{M}$ be a simple NTM accepting L on a work space $\le_a \beta \in B$. Since β can supposed to be time-constructible, the reducing function, $f_{\mathfrak{M}}(w) = \langle w, \mathrm{code}(\mathfrak{M})^{\beta(|w|)}\rangle$, is computable in time $\mathcal{O}(\beta(|w|))$. □

Since, for regular B, the space classes [N]Space(B) are closed under $\le_B$, analogously to Proposition 3 it follows:

Proposition 14. *If* $B_1 \le_a B_2 \le_a B_3$ *for regular sets of bounds* B_1, B_2 *and* B_3*, then*

$$\mathrm{NSpace}(B_1) \subseteq \mathrm{Space}(B_2) \quad \textit{implies} \quad \mathrm{NSpace}(B_3) = \mathrm{Space}(B_3). \quad \square$$

For example, if NLBA $\subseteq$ Space(B_{qlin}), then NSpace(B_{qlin}) = Space(B_{qlin}).

A more interesting application of VS is the following:

Proposition 15. *Let* $A \subseteq \mathbb{X}^*$ *be* NLBA*-complete under linear-time reducibility. Then*

$$\mathrm{Lin}^A = \mathrm{NLin}^A = \mathrm{NLBA}.$$

NLBA $\subseteq \mathrm{Lin}^A$ holds, since any $L \in$ NLBA can be reduced to A by a linear-time computable function. Conversely, if $L \in \mathrm{NLin}^A$ and $A \in$ NLBA, then by [7,14] the complement $\overline{A} = \mathbb{X}^* \setminus A$ belongs to NLBA, too. Now L can be accepted by a nondeterministic linear bounded automaton that simulates the linear-time acception, but replaces each oracle query by a subroutine that guesses an answer and confirms this by means of the acception of A and $\overline{A}$, respectively. Hence $L \in$ NLBA. □

By means of Corollary 1, from Proposition 15 it follows:

Corollary 3. *If the language A is* NLBA*-complete under linear-time reducibility, then* $\mathrm{NTime}^A(B) = \mathrm{Time}^A(B)$ *for any regular set of bounds B.* □

Usually, as oracle sets A with $\mathrm{P}^A = \mathrm{NP}^A$, one takes PSPACE-complete languages A. Corollary 3 shows that our $VS \in$ NLBA can also be used to this purpose.

We conclude with an application concerning the tape-number hierarchy of deterministic linear-time TMs. By [3], "it is plausible that a $k+1$-tape deterministic linear time Turing machine can accept sets not accepted by any k-tape such machine". The following proposition shows that this assertion does not relativize.

Proposition 16. *Let A be* NLBA*-complete under linear-time reducibility. Then there is a number $k_0 \in \mathbb{N}$ such that every language $L \in \mathrm{Lin}^A$ can be accepted by a linear time-bounded deterministic k_0-tape OTM with the oracle A.*

By Proposition 6, V^A is NLin^A-complete. Thus, for any $L \in \mathrm{Lin}^A$ we have $L \leq_{B_{\mathrm{lin}}} V^A$, and the proof of Proposition 6 shows that the reducing word function $f_{\mathfrak{M}}$ can be computed by a deterministic TM $\mathfrak{M}_0$ without any work tape, i.e., with an input tape and an output tape only. Since $\mathrm{NLin}^A = \mathrm{Lin}^A$, V^A can be accepted in linear time by some $\mathfrak{M}_1^A$, where $\mathfrak{M}_1$ is a deterministic k_A-tape OTM, with some number k_A of work tapes. Hence, the composition of $\mathfrak{M}_0$ and $\mathfrak{M}_1$ leads to a deterministic (k_A+1)-tape OTM that accepts L in linear time with respect to the oracle set A. So $k_0 = k_A + 1$ fulfills the assertion. □

For $A = VS$, we get a still sharper result: If $L \in \mathrm{Lin}^{VS} = \mathrm{NLin}^{VS} = $ NLBA, then $L \leq_{B_{\mathrm{lin}}} VS$ via a function $f_{\mathfrak{M}} : w \mapsto \langle w, \mathrm{code}(\mathfrak{M})^{c\cdot|w|}\rangle$, cf. the proof of Proposition 13. In order to decide L relatively to VS in linear time, let a deterministic OTM, on the input $w \in \mathbb{X}^*$, put here $f_{\mathfrak{M}}(w)$ on the oracle tape and halt with the oracle answer. To do this, no further work tape is needed. Thus, L is accepted by a linear time-bounded deterministic 0-tape OTM with oracle VS.

Acknowledgment

The example of the o-regular set of superpolynomial bounds, B_{spol}, has been noticed by an anonymous referee.

References

[1] R. Book, S. Greibach, Quasirealtime languages, Math. Systems Theory 4 (1970) 87–111.
[2] J.-Y. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sawelson, K. Wagner, G. Wechsung, The Boolean hierarchy I: structural properties, SIAM J. Comput. 17 (1988) 1232–1252.
[3] S. Cook, P. Nguyen, Logical Foundations of Proof Complexity, Cambridge University Press, 2010.
[4] D.-Z. Du, K.-I. Ko, Theory of Computational Complexity, Wiley-Interscience, New York, 2000.
[5] A. Durand, M. More, Non-erasing, counting and majority over the linear time hierarchy, Inform. Comput. 174 (2002) 132–142.
[6] L. Fortnow, The role of relativization in complexity theory, Bull. EATCS 52 (1994) 229–244.
[7] N. Immerman, Nondeterministic space is closed under complementation, SIAM J. Comput. 17 (1988) 935–938.
[8] J. Kadin, The polynomial time hierarchy collapses if the Boolean hierarchy collapses, SIAM J. Comput. 17 (1988) 1263–1282.
[9] A.V. Naik, K.W. Regan, D. Sivakumar, On quasilinear-time complexity theory, Theoret. Comput. Sci. 148 (1995) 325–349.
[10] W. Paul, N. Pippenger, E. Szemerèdi, W.T. Trotter, On determinism versus nondeterminism and related problems, in: Proc. of the 24th FOCS, 1983, pp. 429–438.
[11] C.P. Schnorr, Satisfiability is quasilinear complete in NQL, J. ACM 25 (1978) 136–145.
[12] L. Stockmeyer, A. Mayer, Word problems requiring exponential time, in: Proc. of the 5th STOC, 1973, pp. 1–9.
[13] L. Stockmeyer, The polynomial time hierarchy, Theoret. Comput. Sci. 3 (1977) 1–22.
[14] R. Szelepcsényi, The method of forced enumeration for nondeterministic automata, Acta Inform. 26 (1988) 279–284.
[15] G. Wechsung, Vorlesungen zur Komplexitätstheorie, B.G. Teubner, Stuttgart, 2000.
[16] C. Wrathall, Rudimentary predicates and relative computation, SIAM J. Comput. 7 (1978) 194–209.