IFIG
RESEARCH
REPORT

INSTITUT FÜR INFORMATIK

BELOW LINEAR-TIME: DIMENSIONS VERSUS TIME

Martin Kutrib

IFIG RESEARCH REPORT 0005
NOVEMBER 2000

Institut für Informatik
JLU Gießen
Arndtstraße 2
D-35392 Giessen, Germany
Tel: +49-641-99-32141
Fax: +49-641-99-32149
mail@informatik.uni-giessen.de
www.informatik.uni-giessen.de

JUSTUS-LIEBIG-
UNIVERSITÄT
GIESSEN

IFIG Research Report
IFIG Research Report 0005, November 2000

Below Linear-Time: Dimensions versus Time

Martin Kutrib[1]

Institute of Informatics, University of Giessen
Arndtstr. 2, D-35392 Giessen, Germany

Abstract. Deterministic d-dimensional Turing machines are considered. We investigate the classes of languages acceptable by such devices with time bounds of the form $id + r$ where $r \in o(id)$ is a sublinear function. It is shown that for any dimension $d \geq 1$ there exist infinite time hierarchies of separated complexity classes in that range. Moreover, for the corresponding time bounds separated dimension hierarchies are proved.

CR Subject Classification (1998): F.1.3, F.1.1, F.4.3

[1]E-mail: kutrib@informatik.uni-giessen.de

1 Introduction

In the sequel we are concerned with Turing machine computations with time bounds of the form $id+r$ where id denotes the identity function on integers and $r \in o(id)$ a sublinear function. Most of the previous investigations in this area have been done in terms of one-dimensional real-time and linear-time Turing machines.

For the time bounds in question nondeterministic Turing machines would not be fruitful devices for investigations. From [6] we know that the real-time and linear-time classes are identical for one-tape machines $\mathsf{NTIME}_1(id) = \mathsf{NTIME}_1(\mathrm{LIN})$. In [2] it has been shown that the complexity class Q which is defined by nondeterministic multitape real-time computations ($\mathsf{NTIME}(id)$) is equal to the corresponding linear-time languages ($\mathsf{NTIME}(\mathrm{LIN})$). Moreover, it has been shown that two working tapes and a one-way input tape are sufficient to accept the languages from Q in real-time. Thus, for almost all nondeterministic Turing machines there is no difference between real-time and linear-time.

The same does not hold true for deterministic machines. Though in [6] for one tape the identity $\mathsf{DTIME}_1(id) = \mathsf{DTIME}_1(\mathrm{LIN})$ has been proved, for a total of at least two tapes the real-time languages are strictly included in the linear-time languages. Consequently, the investigations have to be in terms of deterministic Turing machines.

Another aspect that, at first glance, might attack the time range of interest is a possible speed-up. The well-known linear speed-up [5] from $t(n)$ to $id + \varepsilon \cdot t(n)$ for arbitrary $\varepsilon > 0$ yields complexity classes close to real-time (i.e. $\mathsf{DTIME}(\mathrm{LIN}) = \mathsf{DTIME}((1+\varepsilon) \cdot id))$ for k-tape and multitape machines but does not allow assertions on the range between real-time and linear-time. An application to the time bound $id + r$, $r \in o(id)$, would result in a slow-down to $id + \varepsilon \cdot (id + r) \geq id + \varepsilon \cdot id$.

Let us recall known time hierarchy results. For a number of $k \geq 2$ tapes in [4, 10] the hierarchy $\mathsf{DTIME}_k(t') \subset \mathsf{DTIME}_k(t)$, if $t' \in o(t)$ and t is time-constructible, has been shown. By the linear speed-up we obtain the necessity of the condition $t' \in o(t)$. The necessity of the constructibility property of t follows from the well-known gap theorem.

Since in case of multitape machines one needs to construct a Turing machine with a fixed number of tapes that simulates machines even with more tapes, the proof of a corresponding hierarchy involves a reduction of the number of tapes. This costs a factor log for the time complexity. The hierarchy $\mathsf{DTIME}(t') \subset \mathsf{DTIME}(t)$, if $t' \cdot \log(t') \in o(t)$ and t is time-constructible, has been proved in [5].

Due to the necessary condition $t' \in o(t)$ resp. $t' \cdot \log(t') \in o(t)$, again, the range between real-time and linear-time is not affected by the known time hierarchy results. On the other hand, it follows immediately from the condition $t' \in o(t)$ and the linear speed-up that there are no infinite hierarchies for time bounds of the form $t + r$, $r \in o(id)$, if $t \geq c \cdot id$, $c > 1$.

Related work concerning higher dimensional Turing machines can be found e.g. in [7] where under the different constraint of on-line computations the trade-off between time and dimensionality is investigated. Upper bounds for the reduction of the dimensions are dealt with e.g. in [9, 11, 12, 14].

Here, on one hand, we are going to present time hierarchies below linear-time for any dimension. On the other hand, dimension hierarchies are presented for every time bound in the range in question. Thus, we obtain a two-dimensional time-dimension hierarchy.

The basic notions and a preliminary result of a technical flavor are the objects of the next section. Section 3 is devoted to the hierarchies below linear-time. In particular, by generalizing a well-known equivalence relation to time complexities above real-time it is shown that specific languages which are constructed dependent on the given time complexity are not acceptable by d-dimensional multitape Turing machines obeying the smaller time bound. Conversely, it is proved by construction that these languages are acceptable by d-dimensional Turing machines whereby the larger time bound is obeyed. In Section 4 the dimension hierarchies are proved by similar witness languages and the same method.

2 Preliminaries

We denote the rational numbers by $\mathbb{Q}$, the integers by $\mathbb{Z}$, the positive integers $\{1, 2, ...\}$ by $\mathbb{N}$ and the set $\mathbb{N} \cup \{0\}$ by $\mathbb{N}_0$. The empty word is denoted by λ and the reversal of a word w by w^R. For the length of w we write $|w|$. We use $\subseteq$ for inclusions and $\subset$ if the inclusion is strict. Let $e_i = (0, \dots, 0, 1, 0, \dots, 0)$ (the 1 is at position i) denote the ith d-dimensional unit vector, then we define $E_d = \{0\} \cup \{e_i \mid 1 \le i \le d\} \cup \{-e_i \mid 1 \le i \le d\}$. For a function $f : \mathbb{N}_0 \to \mathbb{N}$ we denote its i-fold composition by $f^{[i]}$, $i \in \mathbb{N}$. If f is increasing then its inverse is defined according to $f^{-1}(n) = \min\{m \in \mathbb{N} \mid f(m) \ge n\}$. The identity function $n \mapsto n$ is denoted by id. As usual we define the set of functions that grow strictly less than f by $o(f) = \{g : \mathbb{N}_0 \to \mathbb{N} \mid \lim_{n\to\infty} \frac{g(n)}{f(n)} = 0\}$. In terms of orders of magnitude f is an upper bound of the set $O(f) = \{g : \mathbb{N}_0 \to \mathbb{N} \mid \exists\, n_0, c \in \mathbb{N} : \forall\, n \ge n_0 : g(n) \le c \cdot f(n)\}$. Conversely, f is a lower bound of the set $\Omega(f) = \{g : \mathbb{N}_0 \to \mathbb{N} \mid f \in O(g)\}$.

A d-dimensional Turing machine with $k \in \mathbb{N}$ tapes consists of a finite-state control, a read-only one-dimensional one-way input tape and k infinite d-dimensional working tapes. On each tape a read-write head is positioned. At the outset of a computation the Turing machine is in the designated initial state and the input is the inscription of the input tape, all the other tapes are blank. The read-write head of the input tape scans the leftmost symbol of the input whereas all the other heads are positioned on arbitrary tape cells. Dependent on the current state and the currently scanned symbols on the $k + 1$ tapes, the Turing machine changes its state, rewrites the symbols at the head positions of the working tapes and possibly moves the heads independently to a neighboring cell. The head of the input tape may only be moved to the right. With an eye towards language recognition the machines have no extra output tape but the states are partitioned in accepting and rejecting states. More formally:

Definition 1 *A deterministic* d-dimensional Turing machine *with* $k \in \mathbb{N}$ *tapes* (DTM_k^d) *is a system* $\langle S, T, A, \delta, s_0, F \rangle$, *where*

1. S *is the finite set of* internal states,
2. T *is the finite set of* tape symbols *containing the* blank *symbol* $\sqcup$,
3. $A \subseteq T$ *is the set of* input symbols,
4. $s_0 \in S$ *is the* initial state,
5. $F \subseteq S$ *is the set of* accepting states,
6. $\delta : S \times (A \cup \{\sqcup\}) \times T^k \to S \times T^k \times \{0,1\} \times E_d^k$ *is the* partial transition function.

Since the input tape cannot be rewritten we need no new symbol for its current tape cell. Due to the same fact δ may only expect symbols from $A \cup \{\sqcup\}$ on the input tape. The set of rejecting states is implicitly given by the partitioning, i.e. $S \setminus F$. The unit vectors correspond to the possible moves of the read-write heads.

If the set of tape symbols is a Cartesian product of some smaller sets $T = T_1 \times T_2 \times \cdots \times T_l$ we will use the notion *register* for the single parts of a symbol. The concatenation of a register of all tape cells of a tape forms a *track*.

Let $\mathcal{M}$ be a DTM_k^d. A *configuration* of $\mathcal{M}$ at some time $t \geq 0$ is a description of its global state which is a $(2(k+1)+1)$-tuple $(s, f_0, f_1, \ldots, f_k, p_0, p_1, \ldots, p_k)$ where $s \in S$ is the current state, $f_0 : \mathbb{Z} \to A$ and $f_i : \mathbb{Z}^d \to T$ are functions that map the tape cells of the corresponding tape to their current contents, and $p_0 \in \mathbb{Z}$ and $p_i \in \mathbb{Z}^d$ are the current head positions, $1 \leq i \leq k$.

The initial configuration $(s_0, f_0, f_1, \ldots, f_k, 1, 0, \ldots, 0)$ at time 0 is defined by the input word $w = a_1 \cdots a_n \in A^*$, the initial state s_0 and blank working tapes:

$$f_0(m) = \begin{cases} a_m & \text{if } 1 \leq m \leq n \\ \sqcup & \text{otherwise} \end{cases}$$

$$f_i(m_1, \ldots, m_d) = \sqcup \quad \text{for } 1 \leq i \leq k$$

Subsequent configurations are computed according to the *global transition function* Δ: Let $(s, f_0, f_1, \ldots, f_k, p_0, p_1, \ldots, p_k)$ be a configuration and

$$\delta(s, f_0(p_0), f_1(p_1), \ldots, f_k(p_k)) \text{ defined to be } (\tilde{s}, x_1, \ldots, x_k, j_0, j_1, \ldots, j_k).$$

Then the successor configuration is as follows, $1 \le i \le k$:

$$(s', f_0, f_1', \ldots, f_k', p_0', p_1', \ldots, p_k') = \Delta\big((s, f_0, f_1, \ldots, f_k, p_0, p_1, \ldots, p_k)\big) \iff$$

$$s' = \tilde{s}$$

$$f_i'(m_1, \ldots, m_d) = \begin{cases} f_i(m_1, \ldots, m_d) & \text{if } (m_1, \ldots, m_d) \neq p_i \\ x_i & \text{if } (m_1, \ldots, m_d) = p_i \end{cases}$$

$$p_i' = p_i + j_i, \quad p_0' = p_0 + j_0$$

Thus, the global transition function Δ is induced by δ.

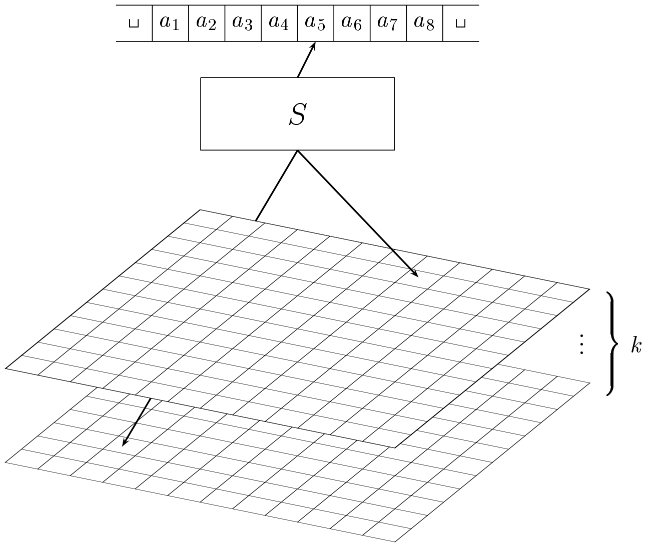

Figure 1: Two-dimensional Turing machine with k working tapes and an input tape.

Throughout the paper we are dealing with so-called multitape machines:

$$\mathrm{DTM}^d = \bigcup_{k \in \mathbb{N}} \mathrm{DTM}_k^d$$

A Turing machine *halts* iff the transition function is undefined for the current configuration. An input word w is accepted by a Turing machine if the machine halts at some time in an accepting state, otherwise it is rejected.

Definition 2 *Let $\mathcal{M} = \langle S, T, A, \delta, s_0, F \rangle$ be a Turing machine.*

1. *A word $w \in A^*$ is* accepted *by $\mathcal{M}$ if $\mathcal{M}$ on input w halts at some time in an accepting state.*
2. *$L(\mathcal{M}) = \{w \in A^* \mid w$ is accepted by $\mathcal{M}\}$ is the* language accepted *by $\mathcal{M}$.*
3. *Let $t : \mathbb{N}_0 \to \mathbb{N}$, $t(n) \ge n + 1$, be a function. A Turing machine is said to be* t-time-bounded *or of* time complexity t *iff it halts on every input of length n after at most $t(n)$ time steps.*

The family of all languages which can be accepted by DTM_k^d with time complexity t is denoted by $\mathsf{DTIME}_k^d(t)$. For multitape machines it holds

$$\mathsf{DTIME}^d(t) = \bigcup_{k \in \mathbb{N}} \mathsf{DTIME}_k^d(t)$$

If t equals the function $id + 1$ acceptance is said to be in *real-time*. The *linear-time* languages are defined according to

$$\mathsf{DTIME}_k^d(\mathrm{LIN}) = \bigcup_{c \in \mathbb{Q}, c \geq 1} \mathsf{DTIME}_k^d(c \cdot id)$$

Since time complexities are mappings to positive integers and have to be greater than or equal to $id + 1$, actually, $c \cdot id$ means $\max\{\lceil c \cdot id \rceil, id + 1\}$. But for convenience we simplify the notation in the sequel.

In order to prove tight time hierarchies in almost all cases honest time bounding functions are required. Usually the notion "honest" is concretized in terms of computability or constructibility of the functions with respect to the device in question.

Definition 3 *Let $d \in \mathbb{N}$ be a constant. A function $f : \mathbb{N}_0 \to \mathbb{N}$ is said to be* DTM^d-time-constructible *iff there exists a* DTM^d *which for every $n \in \mathbb{N}$ on input 1^n halts after exactly $f(n)$ time steps.*

Another common definition of time-constructibility demands the existence of an $O(f)$-time-bounded Turing machine that computes the binary representation of the value $f(n)$ on input 1^n. Both definitions have been proven to be equivalent for multitape machines [8].

The following definition summarizes the properties of honest functions and names them.

Definition 4

1. *The set of all increasing, unbounded* DTM^d*-time-constructible functions f with the property $O(f(n)) \leq f(O(n))$ is denoted by $\mathscr{T}(\mathrm{DTM}^d)$.*
2. *The set of their inverses is $\mathscr{T}^{-1}(\mathrm{DTM}^d) = \{f^{-1} \mid f \in \mathscr{T}(\mathrm{DTM}^d)\}$.*

The properties increasing and unbounded are straightforward. At first glance the property $O(f(n)) \leq f(O(n))$ seems to be restrictive, but it is not. It is easily verified that almost all of the commonly considered time complexities have this property. As usual here we remark that even the family $\mathscr{T}(\mathrm{DTM}^1)$ is very rich. More details can be found for example in [1, 15].

Due to the small time bounds the devices under investigation are too weak for diagonalization. In order to separate complexity classes counting arguments are used. The following equivalence relation is well-known. At least implicitly it has been used several times in connection with real-time computations, e.g. in [5, 13] for Turing machines and in [3] for iterative arrays.

Definition 5 *Let $L \subseteq A^*$ be a language over an alphabet A and $l \in \mathbb{N}_0$ be a constant.*

1. *Two words w and w' are l-equivalent with respect to L if*

$$ww_l \in L \iff w'w_l \in L \text{ for all } w_l \in A^l$$

2. *$N(n, l, L)$ denotes the number of l-equivalence classes of words of length $n - l$ with respect to L (i.e. $|ww_l| = n$).*

The underlying idea is to bound the number of distinguishable equivalence classes. The following lemma gives a necessary condition for a language to be $(id + r)$-time acceptable by a DTM^d.

Lemma 6 *Let $r : \mathbb{N}_0 \to \mathbb{N}$ be a function and $d \in \mathbb{N}$ be a constant. If $L \in \mathsf{DTIME}^d(id + r)$ then there exists a constant $p \in \mathbb{N}$ such that*

$$N(n, l, L) \leq p^{(l+r(n))^d}$$

Proof. Let $\mathcal{M} = \langle S, T, A, \delta, s_0, F\rangle$ be a $(id + r)$-time DTM^d that accepts a language L.

In order to determine an upper bound for the number of l-equivalence classes we consider the possible situations of $\mathcal{M}$ after reading all but l input symbols. The remaining computation depends on the current internal state and the contents of the at most $(2(l+r(n))+1)^d$ cells on each tape that are still reachable during the last at most $l + r(n)$ time steps.

Let $p_1 = \max\{|T|, |S|\}$.

For the $(2(l + r(n)) + 1)^d$ cells per tape there are at most $p_1^{(2(l+r(n))+1)^d}$ different inscriptions. For some $k \in \mathbb{N}$ tapes we obtain altogether at most $p_1^{k(2(l+r(n))+1)^d+1}$ different situations what bounds the number of l-equivalence classes. The lemma follows for $p = p_1^{(k+1)\cdot 3^d}$.

□

3 The Time Hierarchies

In this section we will present the time hierarchies between real-time and linear-time for any dimension $d \in \mathbb{N}$.

Theorem 7 *Let $r : \mathbb{N}_0 \to \mathbb{N}$ and $r' : \mathbb{N}_0 \to \mathbb{N}$ be two increasing functions and $d \in \mathbb{N}$ be a constant. If $r \in \mathscr{T}^{-1}(\mathrm{DTM}^d)$, $r \in O(id^{\frac{1}{d}})$ and $r' \in o(r)$ if $d = 1$ or $r' \in o(r^{1-\varepsilon})$ for an arbitrarily small $\varepsilon > 0$ if $d > 1$ then*

$$\mathsf{DTIME}^d(id + r') \subset \mathsf{DTIME}^d(id + r)$$

Proof. At first let us adjust a constant q dependent on the ε. Choose q such that

$$\frac{d-1}{d^q+d} \leq \varepsilon$$

for $d > 1$ and $q = 1$ for $d = 1$.

Since $r \in \mathscr{T}^{-1}(\mathrm{DTM}^d)$ there exists the function $r^{-1} \in \mathscr{T}(\mathrm{DTM}^d)$.

Now we are prepared to define a witness language L_1 for the assertion.

The words of L_1 are of the form

$$\mathtt{a}^l \mathtt{b}^{r^{-1}(l^{1+d^{q-1}})} w_1 \$ w_1^R ¢ w_2 \$ w_2^R ¢ \cdots ¢ w_s \$ w_s^R ¢ d_1 \cdots d_m y$$

where $l \in \mathbb{N}$ is a positive integer, $s = l^{d^q}$, $m = (d-1) \cdot l^{d^{q-1}}$, $y, w_i \in \{\mathtt{0},\mathtt{1}\}^l$, $1 \leq i \leq s$, and $d_i \in E_{d-1}$, $1 \leq i \leq m$.

The acceptance of such a word is best described by the behavior of an accepting DTM^d $\mathcal{M}$.

During a first phase $\mathcal{M}$ reads $\mathtt{a}^l$ and stores it on a tape. Since d and q are constants $f(l) = l^{1+d^{q-1}}$ is a polynomial and, thus, time-constructible. r^{-1} is constructible per assumption. The time-constructible functions are closed under composition. Therefore, during a second phase $\mathcal{M}$ can simulate a time-constructor for $r^{-1}(f)$ on the stored input $\mathtt{a}^l$ and verify the number of $\mathtt{b}$'s.

Parallel to what follows $\mathcal{M}$ verifies the lengths of the w_i to be l (with the help of the stored $\mathtt{a}^l$) and the numbers s and m ($s = l^{d^q}$ as well as $m = (d-1) \cdot l^{d^{q-1}}$ are time-constructible functions).

When the w_1 appears in the input $\mathcal{M}$ begins to store the subwords w_i in a d-dimensional area of size $l^{d^{q-1}} \times \cdots \times l^{d^{q-1}} \times l^{1+d^{q-1}}$. If, for example, the head of the corresponding tape is located at coordinates $(m_1, \ldots, m_d)$ then the following subword w_i is stored into the cells

$$(m_1, \ldots, m_{d-1}, m_d), (m_1, \ldots, m_{d-1}, m_d + 1), \ldots, (m_1, \ldots, m_{d-1}, m_d + l - 1)$$

Temporarily, w_i is also stored on another tape. Now $\mathcal{M}$ decides where to store the next subword w_{i+1} (for this purpose it simulates appropriate time-constructors for $l^{d^{q-1}}$). Dependent on whether one of the first $d-1$ or the dth coordinate has to be changed $\mathcal{M}$ moves its head back to position $(m_1, \ldots, m_d)$ or keeps its head on position $(m_1, \ldots, m_d + l)$ while reading w_i^R. In both cases w_i^R is verified with the temporarily stored w_i. While reading the following symbol ¢ the head changes to the new coordinates.

The last phase leads to acceptance or rejection. After storing all subwords w_i the last coordinate of the head position is $l^{1+d^{q-1}}$. While reading the d_i $\mathcal{M}$ changes its head simply by adding d_i to the current position. Since $d_i \in E_{d-1}$ the dth coordinate is not affected. This phase leads to a head position $(m_1, \ldots, m_{d-1}, l^{1+d^{q-1}})$. Now the subword y is read and stored on another tape. Finally, $\mathcal{M}$ verifies whether or not y matches one of the subwords which have been stored into the cells

$$(m_1, \ldots, m_{d-1}, 0), \ldots, (m_1, \ldots, m_{d-1}, l^{1+d^{q-1}} - 1)$$

(if there are stored subwords in these cells at all). $\mathcal{M}$ accepts if and only if it finds a matching subword.

Altogether, $\mathcal{M}$ needs n time steps for reading the whole input and at most another $l^{1+d^{q-1}}$ time steps for comparing the y with the stored subwords. The first part of the input contains $r^{-1}(l^{1+d^{q-1}})$ symbols $\mathtt{b}$. Therefore, $n > r^{-1}(l^{1+d^{q-1}})$ and since r is increasing $r(n) \geq r(r^{-1}(l^{1+d^{q-1}})) = l^{1+d^{q-1}}$. We conclude that $\mathcal{M}$ obeys the time complexity $id + r$ and, hence, $L_1 \in \mathsf{DTIME}^d(id + r)$.

Assume now L_1 is acceptable by some DTM^d $\mathcal{M}$ with time complexity $id + r'$.

Two words

$$\mathtt{a}^l\mathtt{b}^{r^{-1}(l^{1+d^{q-1}})}w_1\$w_1^R¢w_2\$w_2^R¢\cdots¢w_s\$w_s^R¢$$

and

$$\mathtt{a}^l\mathtt{b}^{r^{-1}(l^{1+d^{q-1}})}w'_1\$w_1'^R¢w'_2\$w_2'^R¢\cdots¢w'_s\$w_s'^R¢$$

are not $(m + l)$-equivalent with respect to L_1 if the sets $\{w_1, \ldots, w_s\}$ and $\{w'_1, \ldots, w'_s\}$ are not equal. There are exactly $\binom{2^l}{l^{d^q}}$ different subsets of $\{\mathtt{0}, \mathtt{1}\}^l$ with l^{d^q} elements. It follows:

$$\begin{aligned} N(n, l+m, L_1) &\geq \binom{2^l}{l^{d^q}} > \left(\frac{2^l - l^{d^q}}{l^{d^q}}\right)^{l^{d^q}} \\ &\geq \left(\frac{2^{\frac{l}{2}}}{l^{d^q}}\right)^{l^{d^q}} = \left(2^{\frac{l}{2}-\log(l^{d^q})}\right)^{l^{d^q}} \\ &\geq \left(2^{\Omega(l)}\right)^{l^{d^q}} = 2^{\Omega(l^{1+d^q})} \end{aligned}$$

for all sufficiently large l.

On the other hand, by Lemma 6 the number of equivalence classes distinguishable by $\mathcal{M}$ is bounded for a constant $p \in \mathbb{N}$:

$$N(n, l+m, L_1) \leq p^{(l+m+r'(n))^d}$$

For n we have

$$\begin{aligned} n &= l + r^{-1}(l^{1+d^{q-1}}) + (2l+2) \cdot l^{d^q} + (d-1) \cdot l^{d^{q-1}} + l \\ &= O(l^{1+d^q}) + r^{-1}(l^{1+d^{q-1}}) \end{aligned}$$

Since $r \in O(id^{\frac{1}{d}})$ it follows $r^{-1} \in \Omega(id^d)$. Therefore,

$$r^{-1}(l^{1+d^{q-1}}) \in \Omega(l^{d+d^q})$$

We conclude

$$n \leq c_1 \cdot r^{-1}(l^{1+d^{q-1}}) \text{ for some } c_1 \in \mathbb{N}$$

Due to the property $O(r^{-1}(n)) \leq r^{-1}(O(n))$ we obtain

$$n \leq r^{-1}(c_2 \cdot l^{1+d^{q-1}}) \text{ for some } c_2 \in \mathbb{N}$$

From $1-\varepsilon \leq 1-\frac{d-1}{d^q+d} = \frac{d^q+1}{d^q+d} = \frac{d^{q-1}+\frac{1}{d}}{d^{q-1}+1}$ and $r' \in o(r^{1-\varepsilon})$ it follows

$$\begin{aligned} r'(n) &\leq r'(r^{-1}(c_2 \cdot l^{1+d^{q-1}})) \\ &\leq o(r(r^{-1}(c_2 \cdot l^{1+d^{q-1}}))^{\frac{d^{q-1}+\frac{1}{d}}{d^{q-1}+1}}) \\ &= o(l^{\frac{1}{d}+d^{q-1}}) \end{aligned}$$

By $l+m = l+(d-1)\cdot l^{d^{q-1}} = O(l^{d^{q-1}})$ it holds

$$\begin{aligned} (l+m+r'(n))^d &= (O(l^{d^{q-1}}) + o(l^{\frac{1}{d}+d^{q-1}}))^d \\ &= o(l^{\frac{1}{d}+d^{q-1}})^d = o(l^{1+d^q}) \end{aligned}$$

Finally, the number of distinguishable equivalence classes is

$$N(n, l+m, L_1) \leq p^{o(l^{1+d^q})} = 2^{o(l^{1+d^q})}$$

Now we have the contradiction that previously $N(n, l+m, L_1)$ has been calculated to be at least $2^{\Omega(l^{1+d^q})}$ what proves $L_1 \notin \mathsf{DTIME}^d(id+r')$. □

For one-dimensional machines we have hierarchies from real-time to linear-time. Due to the possible speed-up from $id+r$ to $id+\varepsilon\cdot r$ the condition $r' \in o(r)$ cannot be relaxed. Example functions for every dimension are $id^{\frac{1}{i}}$ and $\log^{[i]}$ (cf. Example 10).

4 The Dimension Hierarchies

By a similar witness language and the same method infinite dimension hierarchies for the time complexities in question can be shown.

Theorem 8 *Let $r : \mathbb{N}_0 \to \mathbb{N}$ be an increasing function and $d \in \mathbb{N}$ be a constant. If $r \in o(id^{\frac{1}{d}})$ then*

$$\mathsf{DTIME}^{d+1}(id) \setminus \mathsf{DTIME}^d(id+r) \neq \emptyset$$

Proof. The words of the witness language L_2 are of the form

$$w_1 \$ w_1^R ¢ w_2 \$ w_2^R ¢ \cdots ¢ w_s \$ w_s^R ¢ d_1 \cdots d_m y$$

where $l \in \mathbb{N}$ is a positive integer, $s = l^d$, $m = d \cdot l$, $y, w_i \in \{0,1\}^l$, $1 \leq i \leq s$, and $d_i \in E_d$, $1 \leq i \leq m$.

An accepting $(d+1)$-dimensional real-time machine $\mathcal{M}$ works as follows. The subwords w_i are stored into a $(d+1)$-dimensional area of size $l \times l \times \cdots \times l$. The first symbols of the w_i are stored at the l^d positions

$$(0,0,\ldots,0) \text{ to } (l-1, l-1, \ldots, l-1, 0)$$

The words itself are stored along the $(d+1)$th dimension.

After storing the subwords $\mathcal{M}$ moves its corresponding head as requested by the d_i. Since the d_i are belonging to E_d this movement is within the first d dimensions only. Finally, when the y appears in the input $\mathcal{M}$ tries to compare the y with the subword stored at the current position. $\mathcal{M}$ accepts if a subword has been stored at the current position at all and if the subword matches the y. Thus, $L_2 \in \mathsf{DTIME}^{d+1}(id+1)$.

In order to apply Lemma 6 we observe that, again, two words

$$w_1\$w_1^R¢w_2\$w_2^R¢\cdots¢w_s\$w_s^R¢$$

and

$$w_1'\$w_1'^R¢w_2\$w_2'^R¢\cdots¢w_s\$w_s'^R¢$$

are not $(m+l)$-equivalent with respect to L_2 if the sets $\{w_1,\dots,w_s\}$ and $\{w_1',\dots,w_s'\}$ are not equal. Therefore, L_2 induces at least

$$N(n,l+m,L_2) \geq \binom{2^l}{l^d} \geq 2^{\Omega(l^{d+1})}$$

equivalence classes for all sufficiently large l.

On the other hand, we obtain an upper bound of the number of distinguishable equivalence classes for an $(id+r)$-time DTM^d $\mathcal{M}$ as follows

$$\begin{aligned}
N(n,l+m,L_2) &\leq p^{(l+m+r(n))^d} \\
&= p^{(l+d\cdot l+r((2l+2)\cdot l^d+l+d\cdot l))^d} \\
&\leq p^{(O(l)+r(c_1\cdot l^{d+1}))^d} \text{ for some } c_1 \in \mathbb{N} \\
&\leq p^{(O(l)+o(c_1\cdot l^{d+1})^{\frac{1}{d}})^d} \\
&= p^{(O(l)+o(l^{\frac{d+1}{d}}))^d} \\
&= p^{o(l^{\frac{d+1}{d}})^d} \\
&= p^{o(l^{d+1})} = 2^{o(l^{d+1})}
\end{aligned}$$

From the contradiction $L_2 \notin \mathsf{DTIME}^d(id+r)$ follows. □

The inclusions $\mathsf{DTIME}^{d+1}(id) \subseteq \mathsf{DTIME}^{d+1}(id+r)$ and $\mathsf{DTIME}^{d}(id+r) \subseteq \mathsf{DTIME}^{d+1}(id+r)$ are trivial. An application of Theorem 8 yields the hierarchies:

Corollary 9 *Let $r : \mathbb{N}_0 \to \mathbb{N}$ be an increasing function and $d \in \mathbb{N}$ be a constant. If $r \in o(id^{\frac{1}{d}})$ then*

$$\mathsf{DTIME}^d(id+r) \subset \mathsf{DTIME}^{d+1}(id+r)$$

Note that despite the condition $r \in o(id^{\frac{1}{d}})$ the dimension hierarchies can touch $r = id^{\frac{1}{d}}$:

$$id^{\frac{1}{d}} \in o(id^{\frac{1}{d-1}}) \text{ and } \mathsf{DTIME}^{d-1}(id+id^{\frac{1}{d}}) \subset \mathsf{DTIME}^{d}(id+id^{\frac{1}{d}})$$

The following example is based on natural functions. It combines both types of hierarchies.

Example 10 Since $\mathscr{T}(\mathrm{DTM}^d)$ is closed under composition and contains 2^{id} and id^c, $c \geq 1$, the functions $\log^{[i]}$, $i \geq 1$, and $id^{\frac{1}{c}}$ are belonging to $\mathscr{T}^{-1}(\mathrm{DTM}^d)$. (Actually, the inverses of 2^{id} and id^c are $\lceil \log \rceil$ and $\lceil id^{\frac{1}{c}} \rceil$ but as mentioned before we simplify the notation for convenience.)

For $d = 1$ trivially $id^{\frac{1}{i+1}} \in o(id^{\frac{1}{i}})$ and $\log^{[i+1]} \in o(\log^{[i]})$.

For $d > 1$ we need to find an ε such that $id^{\frac{1}{i+1}} \in o(id^{\frac{1}{i}(1-\varepsilon)})$ resp. $\log^{[i+1]} \in o((\log^{[i]})^{1-\varepsilon})$.

In the second case we have $\log(\log^{[i]})$ and $(\log^{[i]})^{1-\varepsilon}$ and, therefore, the condition is fulfilled for all $\varepsilon < 1$.

$$\begin{array}{ccccccccc}
\mathsf{DTIME}(id+id) & & \mathsf{DTIME}^2(id+id^{\frac{1}{2}}) & & \mathsf{DTIME}^3(id+id^{\frac{1}{3}}) & & \mathsf{DTIME}^4(id+id^{\frac{1}{4}}) & & \\
\cup & & \cup & & \cup & & \cup & & \\
\mathsf{DTIME}(id+\log) & \subset & \mathsf{DTIME}^2(id+\log) & \subset & \mathsf{DTIME}^3(id+\log) & \subset & \mathsf{DTIME}^4(id+\log) & \subset & \cdots \\
\cup & & \cup & & \cup & & \cup & & \\
\mathsf{DTIME}(id+\log^{[2]}) & \subset & \mathsf{DTIME}^2(id+\log^{[2]}) & \subset & \mathsf{DTIME}^3(id+\log^{[2]}) & \subset & \mathsf{DTIME}^4(id+\log^{[2]}) & \subset & \cdots \\
\cup & & \cup & & \cup & & \cup & & \\
\mathsf{DTIME}(id+\log^{[3]}) & \subset & \mathsf{DTIME}^2(id+\log^{[3]}) & \subset & \mathsf{DTIME}^3(id+\log^{[3]}) & \subset & \mathsf{DTIME}^4(id+\log^{[3]}) & \subset & \cdots \\
\cup & & \cup & & \cup & & \cup & & \\
\vdots & & \vdots & & \vdots & & \vdots & & \\
\cup & & \cup & & \cup & & \cup & & \\
\mathsf{DTIME}(id) & \subset & \mathsf{DTIME}^2(id) & \subset & \mathsf{DTIME}^3(id) & \subset & \mathsf{DTIME}^4(id) & \subset & \cdots
\end{array}$$

The first case holds if and only if $\frac{1}{i+1} < \frac{1}{i}(1-\varepsilon)$. Thus, if $\frac{i}{i+1} < 1 - \varepsilon$ and therefore, if $\varepsilon < 1 - \frac{i}{i+1}$. We conclude that the condition is fulfilled for all $\varepsilon < \frac{1}{i+1}$.

$$\begin{array}{ccccccccc}
\mathsf{DTIME}(id+id) & & & & & & & & \\
\cup & & & & & & & & \\
\mathsf{DTIME}(id+id^{\frac{1}{2}}) & \subset & \mathsf{DTIME}^2(id+id^{\frac{1}{2}}) & & & & & & \\
\cup & & \cup & & & & & & \\
\mathsf{DTIME}(id+id^{\frac{1}{3}}) & \subset & \mathsf{DTIME}^2(id+id^{\frac{1}{3}}) & \subset & \mathsf{DTIME}^3(id+id^{\frac{1}{3}}) & & & & \\
\cup & & \cup & & \cup & & & & \\
\mathsf{DTIME}(id+id^{\frac{1}{4}}) & \subset & \mathsf{DTIME}^2(id+id^{\frac{1}{4}}) & \subset & \mathsf{DTIME}^3(id+id^{\frac{1}{4}}) & \subset & \mathsf{DTIME}^4(id+id^{\frac{1}{4}}) & & \\
\cup & & \cup & & \cup & & \cup & & \\
\vdots & & \vdots & & \vdots & & \vdots & & \\
\cup & & \cup & & \cup & & \cup & & \\
\mathsf{DTIME}(id) & \subset & \mathsf{DTIME}^2(id) & \subset & \mathsf{DTIME}^3(id) & \subset & \mathsf{DTIME}^4(id) & \subset & \cdots
\end{array}$$

References

[1] Balcázar, J. L., Díaz, J., and Gabarró, J. *Structural Complexity I*. Springer, Berlin, 1988.

[2] Book, R. V. and Greibach, S. A. *Quasi-realtime languages*. Math. Systems Theory 4 (1970), 97–111.

[3] Cole, S. N. *Real-time computation by n-dimensional iterative arrays of finite-state machines*. IEEE Trans. Comput. C-18 (1969), 349–365.

[4] Fürer, M. *The tight deterministic time hierarchy*. Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing (STOC '82), 1982, pp. 8–16.

[5] Hartmanis, J. and Stearns, R. E. *On the computational complexity of algorithms*. Trans. Amer. Math. Soc. 117 (1965), 285–306.

[6] Hennie, F. C. *One-tape, off-line turing machine computations*. Inform. Control 8 (1965), 553–578.

[7] Hennie, F. C. *On-line turing machine computations*. IEEE Trans. Elect. Comput. EC-15 (1966), 35–44.

[8] Kobayashi, K. *On proving time constructibility of functions*. Theoret. Comput. Sci. 35 (1985), 215–225.

[9] Loui, M. C. *Simulations among multidimensional turing machines*. Theoret. Comput. Sci. 21 (1982), 145–161.

[10] Paul, W. J. *On time hierarchies*. J. Comput. System Sci. 19 (1979), 197–202.

[11] Paul, W., Seiferas, J. I., and Simon, J. *An information-theoretic approach to time bounds for on-line computation*. J. Comput. System Sci. 23 (1981), 108–126.

[12] Pippenger, N. and Fischer, M. J. *Relations among complexity measures*. J. Assoc. Comput. Mach. 26 (1979), 361–381.

[13] Rosenberg, A. L. *Real-time definable languages*. J. Assoc. Comput. Mach. 14 (1967), 645–662.

[14] Stoß, H.-J. *Zwei-Band Simulation von Turingmaschinen*. Computing 7 (1971), 222–235.

[15] Wagner, K. and Wechsung, G. *Computational Complexity*. Reidel, Dordrecht, 1986.