Relativizations of the P =? DNP Question for the BSS Model

Christine Gaßner

Institut für Mathematik und Informatik, Ernst-Moritz-Arndt-Universität,
Robert-Blum-Str. 2, 17487 Greifswald, Germany
gassnerc@uni-greifswald.de

Abstract. We consider the uniform BSS model of computation where the machines can perform additions, multiplications, and tests of the form $x \geq 0$. The oracle machines can also check whether a tuple of real numbers belongs to a given oracle set $\mathcal{O}$ or not. We construct oracles such that the classes P and DNP relative to these oracles are equal or not equal.

1 Introduction

The uniform BSS model of computation was introduced in [Blum et al. 1989]. The BSS machines can perform labelled instructions of the form $Z_i := Z_j + Z_k$, $Z_i := Z_j - Z_k$, $Z_i := Z_j \cdot Z_k$, $Z_j := c$, *if* $Z_j \geq 0$ *then goto* l_1 *else goto* l_2, $Z_{I_j} := Z_{I_k}$, $I_j := 1$, $I_j := I_j + 1$, and *if* $I_j = I_k$ *then goto* l_1 *else goto* l_2. Each assignment of an input $(x_1, \ldots, x_n) \in \bigcup_{i \geq 1} \mathbb{R}^i$ to the registers of a machine $\mathcal{M}$ is realized by $Z_1 := x_1; \ldots; Z_n := x_n; I_1 := n; \ldots; I_{k_\mathcal{M}} := n$. Moreover, *oracle machines* can execute *if* $(Z_1, \ldots, Z_{I_1}) \in \mathcal{O}$ *then goto* l_1 *else goto* l_2 for some oracle $\mathcal{O} \subseteq \mathbb{R}^\infty$. The *non-deterministic* machines are able to guess an arbitrary number of arbitrary elements $y_1, \ldots, y_m \in \mathbb{R}$ in one step after the input and to assign the guesses to $Z_{I_1+1}, \ldots, Z_{I_1+m}$. A (digital) non-deterministic BSS machine $\mathcal{M}$ *accepts* an input $(x_1, \ldots, x_n) \in \mathbb{R}^\infty$ if there is some guessed sequence $(y_1, \ldots, y_m) \in \mathbb{R}^\infty$ and $(y_1, \ldots, y_m) \in \{0,1\}^\infty$, respectively, such that $\mathcal{M}$ outputs 1 on input $(x_1, \ldots, x_n)$ for the guesses $y_1, \ldots, y_m$. Let $\mathrm{P}_\mathbb{R}$, $\mathrm{DNP}_\mathbb{R}$, and $\mathrm{NP}_\mathbb{R}$ be the classes of problems recognized by deterministic, digital non-deterministic, and non-deterministic machines, respectively, in polynomial time. Let $\mathrm{P}^\mathcal{O}_\mathbb{R}$, $\mathrm{DNP}^\mathcal{O}_\mathbb{R}$, and $\mathrm{NP}^\mathcal{O}_\mathbb{R}$ are the corresponding classes for one given oracle $\mathcal{O}$. We have $\mathrm{P}_\mathbb{R} \subseteq \mathrm{DNP}_\mathbb{R} \subseteq \mathrm{NP}_\mathbb{R}$ and $\mathrm{P}^\mathcal{O}_\mathbb{R} \subseteq \mathrm{DNP}^\mathcal{O}_\mathbb{R} \subseteq \mathrm{NP}^\mathcal{O}_\mathbb{R}$.

In [Baker et al. 1975] and [Emerson 1994] for Turing machines and the BSS model, respectively, oracles were defined in order to get the equality of relativized versions of $\mathrm{P}_{(\mathbb{R})}$ and $\mathrm{NP}_{(\mathbb{R})}$. Such a *universal oracle* $\mathcal{O}$ can be defined by $\mathcal{O} = \bigcup_{i \geq 1} W_i$ where $W_0 = \emptyset$ and

$$W_i = \{(\underbrace{1, \ldots, 1}_{t\times}, \boldsymbol{x}, \mathit{Code}(\mathcal{M})) \in \mathbb{R}^i \mid$$
$$\mathcal{M} \text{ is a non-deterministic machine using } \textstyle\bigcup_{j<i} W_j \text{ as oracle } \ \& \ \mathcal{M}(\boldsymbol{x}) \downarrow^t\}.$$

Thus, we get the following.

Andrej Bauer, Peter Hertling, Ker-I Ko (Eds.)
6th Int'l Conf. on Computability and Complexity in Analysis, 2009, pp. 141-148
http://drops.dagstuhl.de/opus/volltexte/2009/2266

Proposition 1. *There is an oracle $\mathcal{O}$ such that* $\mathrm{P}_{\mathbb{IR}}^{\mathcal{O}} = \mathrm{DNP}_{\mathbb{IR}}^{\mathcal{O}} = \mathrm{NP}_{\mathbb{IR}}^{\mathcal{O}}$.

In [Emerson 1994] Emerson presented also an oracle $\mathcal{Q}$ such that $\mathrm{P}_{\mathbb{IR}}^{\mathcal{Q}} \neq \mathrm{NP}_{\mathbb{IR}}^{\mathcal{Q}}$. Emerson's proof technique also allows to separate relativized versions of $\mathrm{DNP}_{\mathbb{IR}}$ and $\mathrm{NP}_{\mathbb{IR}}$ by a diagonalization procedure in the following way. Let $U \subseteq \mathbb{IR}^\infty$ be a set of codes $\boldsymbol{u}$ representing all pairs that contain a polynomial $p_{\boldsymbol{u}}$ and the program $P_{\boldsymbol{u}}$ of a digital non-deterministic oracle BSS machine. Let $\mathcal{N}_{\boldsymbol{u}}^{\mathcal{B}}$ be the machine using an oracle $\mathcal{B} \subseteq \mathbb{IR}^\infty$ and performing only $p_{\boldsymbol{u}}(n)$ instructions of P_u on inputs of size n. Let the oracle $\mathcal{Q}_1 = \bigcup_{i\geq 1} W_i$ be defined in stages. Let $V_0 = \emptyset$.

Stage $i \geq 1$:

Let

$K_i = \{\boldsymbol{u} \in U \mid (\forall \mathcal{B} \subseteq \mathbb{IR}^\infty)$
$(\mathcal{N}_{\boldsymbol{u}}^{\mathcal{B}}$ does not use any $r > i$ in a query on input $\boldsymbol{u})\}$,

$W_i = \bigcup_{k<i} V_k$,

$V_i = \{(i+1, \boldsymbol{u}) \mid \boldsymbol{u} \in K_i \;\&\; \mathcal{N}_{\boldsymbol{u}}^{W_i}$ does not accept $\boldsymbol{u}\}$.

The defined sequence of codes, $K_1, K_2, \ldots$, covers the set of all digital non-deterministic oracle machines recognizing problems in $\mathrm{DNP}_{\mathbb{IR}}^{\mathcal{B}}$ for some $\mathcal{B}$. Consequently we get $L_1 = \{\boldsymbol{y} \mid (\exists n \in \mathbb{N}^+)((n, \boldsymbol{y}) \in \mathcal{Q}_1)\} \notin \mathrm{DNP}_{\mathbb{IR}}^{\mathcal{Q}_1}$. On the other hand, we have $L_1 \in \mathrm{NP}_{\mathbb{IR}}^{\mathcal{Q}_1}$ since a non-deterministic BSS machine can guess each integer in one step.

Proposition 2. *There is an oracle $\mathcal{Q}$ such that* $\mathrm{DNP}_{\mathbb{IR}}^{\mathcal{Q}} \neq \mathrm{NP}_{\mathbb{IR}}^{\mathcal{Q}}$.

Moreover, by analogy with [Gaßner 2009] it is also possible to show $\mathrm{DNP}_{\mathbb{IR}}^{\mathbb{Z}} \neq \mathrm{NP}_{\mathbb{IR}}^{\mathbb{Z}}$ and $\mathrm{DNP}_{\mathbb{IR}}^{\mathbb{Q}} \neq \mathrm{NP}_{\mathbb{IR}}^{\mathbb{Q}}$.

It remains to show that there are also oracles such that the classes $\mathrm{P}_{\mathbb{IR}}$ and $\mathrm{DNP}_{\mathbb{IR}}$ relative to these oracles are not equal. For the computation over structures of enumerable signature, a method to separate relativized classes of problems recognized by deterministic and digital non-deterministic machines, respectively, goes back to T. Baker, J. Gill, and R. Solovay [Baker et al. 1975]. In order to obtain the inequality between relativized versions of P_{R} and $\mathrm{DNP}_{\mathsf{R}}$ for oracle machines over the ordered ring $\mathsf{R} = (\mathbb{IR}; 0, 1; +, -, \cdot; \geq)$, we can use the enumerability of all polynomials $p : \mathbb{N} \to \mathbb{N}$ and all programs of deterministic oracle machines and diagonalization techniques by analogy with [Baker et al. 1975]. Let $i \in \mathbb{N}^+$ be the code of the pair containing the i^{th} polynomial p_i and the i^{th} program P_i of a deterministic oracle machine using only the machine constants 0 and 1. If $\mathcal{N}_i^{\mathcal{B}}$ is the machine which uses an oracle $\mathcal{B} \subseteq \mathbb{IR}^\infty$ and performs only $p_i(n)$ instructions of P_i on inputs of size n, then the definition of the oracle $\mathcal{Q}_2 = \bigcup_{i\geq 1} W_i$ is possible in stages as in [Baker et al. 1975]. Let $V_0 = \emptyset$ and $m_0 = 0$.

Stage $i \geq 1$: Let n_i be any integer such that $n_i > m_{i-1}$ and $p_i(n_i) + n_i < 2^{n_i}$. Moreover, let

$W_i = \bigcup_{j<i} V_j$,

$V_i = \{\boldsymbol{x} \in \{0,1\}^{n_i} \mid \mathcal{N}_i^{W_i}$ rejects $(0, \ldots, 0) \in \mathbb{IR}^{n_i}$
$\&\; \boldsymbol{x}$ is not queried by $\mathcal{N}_i^{W_i}$ on input $(0, \ldots, 0) \in \mathbb{IR}^{n_i}\}$,

$m_i = 2^{n_i}$.
For $L_2 = \{\boldsymbol{y} \mid (\exists i \in \mathbb{N}^+)(\boldsymbol{y} \in \mathbb{R}^{n_i} \ \&\ V_i \neq \emptyset)\}$, we get $L_2 \in \mathrm{DNP}_{\mathsf{R}}^{\mathcal{Q}_2} \setminus \mathrm{P}_{\mathsf{R}}^{\mathcal{Q}_2}$.

Proposition 3. *For BSS machines using only the constants* 0 *and* 1*, there is an oracle* $\mathcal{Q}$ *such that* $\mathrm{P}_{\mathsf{R}}^{\mathcal{Q}} \neq \mathrm{DNP}_{\mathsf{R}}^{\mathcal{Q}}$.

This method as well as Emerson's method fail if we want to construct an oracle such that the relativized versions of $\mathrm{P}_{\mathbb{R}}$ and $\mathrm{DNP}_{\mathbb{R}}$ are not equal. We cannot enumerate the programs of all deterministic BSS machines, and the digital nondeterministic machines cannot guess any integer in one step. A discussion about the possibilities to transfer the ideas of [Baker et al. 1975] and [Emerson 1994] was done in [Gaßner 2008] for several types of groups. This discussion gives also insights which constructions can be used for which types of rings. In the next section we want to show that it is still possible to use diagonalization techniques for separating the classes $\mathrm{P}_{\mathbb{R}}$ and $\mathrm{DNP}_{\mathbb{R}}$ relative to an oracle. Our construction requires to consider a sequence of sequences of sets of machines and consequently a new recursive definition in every stage of a recursive definition. Techniques of this kind are often used if more natural decision problems having special properties are not known. For models of computation over algebraic structures, a summary of papers where these techniques have been applied is given, for instance, in [Bürgisser 1999]. In the last section we derive a suitable oracle from the Real Knapsack Problem such that the resulting relativized versions of $\mathrm{P}_{\mathbb{R}}$ and $\mathrm{DNP}_{\mathbb{R}}$ are also not equal. This construction is possible without using the powerful diagonalization techniques.

2 The Separation of Relativized Versions of $\mathrm{P}_{\mathbb{R}}$ and $\mathrm{DNP}_{\mathbb{R}}$ by Diagonalization Techniques

Now let us consider again the BSS machines over $(\mathbb{R}; \mathbb{R}; +, -, \cdot; \geq)$ where any real number can be a machine constant. Since we also want to define an oracle

$$\mathcal{Q}_3 \subseteq \bigcup_{i \geq 1} \mathbb{N}^{n_i}$$

recursively, we will at first define a suitable sequence $((\mathcal{K}_{i,j})_{j \geq 1})_{i \geq 1}$ of sequences containing all deterministic oracle BSS machines working in polynomial time. For any oracle $\mathcal{B} \subseteq \mathbb{R}^\infty$, any deterministic oracle BSS machine $\mathcal{N}^{\mathcal{B},c_1,\ldots,c_k}$ is determined by its machine constants $c_1, \ldots, c_k$ and a program P which is encoded by a tuple in $\{0,1\}^\infty$. Let every character of the program P, including the indices $j \in \{1, \ldots, k\}$ of the constants c_j, be unambiguously translated into a finite sequence in $\{0,1\}^\infty$ and let the oracle queries be encoded independently of the used oracle $\mathcal{B}$ by taking the same sequence of characters 0 and 1 as code for all oracle queries. Consequently, the set Prog of all programs of oracle machines and the set poly of all polynomial functions of $\mathbb{N}$ into $\mathbb{N}$ are enumerable. We will take the positive integers in order

– to enumerate all $(p_1, P_1), (p_2, P_2), (p_3, P_3), \ldots \in \mathsf{poly} \times \mathsf{Prog}$,

– to characterize the behavior of all oracle machines on special inputs of size $n_1, n_2, \ldots$ by additional numbers $N_{\mathrm{char}}(i, c_1, \ldots, c_{k_i})$ which are dependent only on the following properties:
  - The machines perform $p_i(n_i)$ instructions on inputs of size n_i.
  - The machines use only the reals $c_1, \ldots, c_{k_i}$ as machine constants.
  - The inputs belong to $\{0\}^{n_i-1} \times \{N \in \mathbb{N} \mid N \geq N_{\mathrm{char}}(i, c_1, \ldots, c_{k_i})\}$.

The Definition of the Machine $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$. *Any $i \in \mathbb{N}^+$ is the number of a pair $(p_i, P_i) \in \mathsf{poly} \times \mathsf{Prog}$ which determines a class of deterministic oracle machines $\{\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}} \mid \mathcal{B} \subseteq \mathbb{R}^\infty \ \&\ c_1, \ldots, c_{k_i} \in \mathbb{R}\}$ by the following.*

(a) The BSS machine $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ performs the instructions of the program P_i.
(b) If $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ queries an oracle, then $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ uses the oracle $\mathcal{B}$.
(c) The only constants of $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ are $c_1, \ldots, c_{k_i}$ encoded by $1, \ldots, k_i$ in the code of P_i.
(d) The number of the instructions of P_i carried out by $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ is simultaneously counted by $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ by means of an additional index register.
(e) For any input in $\mathbb{R}^n$, the machine $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ halts after at most $p_i(n)$ steps of the execution of P_i. (The bound $p_i(n)$ can be computed by using index registers.)
(f) If the output of P_i is reached in this time, then $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ outputs the value determined by P_i, $\mathcal{B}$, and $c_1, \ldots, c_{k_i}$. If the output instruction of P_i is not reached in this time, then $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ rejects the input.

Then, for any oracle $\mathcal{B} \subseteq \mathbb{R}^\infty$ and any problem $\mathcal{P} \in \mathrm{P}_{\mathbb{R}}^{\mathcal{B}}$ there are an $i \geq 1$ and constants $c_1, \ldots, c_{k_i}$ such that the machine $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ decides $\mathcal{P}$.

Let us now characterize the behavior of $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ on inputs of the form $(0, \ldots, 0, x) \in \mathbb{N}^{n_i}$. The value of any register computed by $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ on these inputs can be described by some term of the form (1). We are especially interested in oracles $\mathcal{B} \subseteq \mathbb{N}^\infty$.

The Definition of the Number $N_{\mathrm{char}}(i, c_1, \ldots, c_{k_i})$. *We consider the sequence $(f_1, f_2, \ldots, f_s)$ containing all polynomials $f_k \in \mathbb{R}[x]$ whose values $f_k(x)$ can be described by the terms of the following form*

$$\sum_{j=0}^{2^{p_i(n_i)}} \Big(\sum_{j_1,\ldots,j_{k_i}=0}^{2^{p_i(n_i)}} \alpha_{j_1,\ldots,j_{k_i},j} c_1^{j_1} \cdots c_{k_i}^{j_{k_i}} \Big) x^j \tag{1}$$

where any $\alpha_{j_1,\ldots,j_{k_i},j} \in \mathbb{Z} \cap [-2^{p_i(n_i)}, 2^{p_i(n_i)}]$. Let $N_{\mathrm{char}}(i, c_1, \ldots, c_{k_i})$ be the Cantor number of $(\mu_1, \ldots, \mu_s, \nu_1, \ldots, \nu_s, \mu, \mu', \nu)$ given by

$$\mu_k = \mathrm{code}(f_k) \in \mathbb{N}^+ \qquad \text{if } f_k \in \mathbb{Q}[x], \tag{2}$$

$$\mu_k = 0 \qquad \text{if } f_k \notin \mathbb{Q}[x], \tag{3}$$

$$\nu_k = \lim_{x\to\infty} \operatorname{sgn}(f_k(x)), \tag{4}$$

$$\mu = \min \bigcap_{\substack{k=1,\ldots,s \\ \text{degree}(f_k)\geq 1}} \{n \in \mathbb{N} \mid \forall x(f_k(x) = 0 \vee f_k(x) = 1 \Rightarrow n > x)\}, \tag{5}$$

$$\mu' = \min \bigcap_{\substack{k=1,\ldots,s \\ \mu_k = 0}} \{n \in \mathbb{N} \mid (\forall x \in \mathbb{N})(f_k(x) \in \mathbb{N} \Rightarrow n > x)\}, \tag{6}$$

$$\nu = \min \bigcap_{k=1,\ldots,s} \{n \in \mathbb{N} \mid f_k(n) < 2^n\}. \tag{7}$$

Remark 1. Here, $\operatorname{sgn}(x) = 1$ iff $x > 0$, $\operatorname{sgn}(x) = -1$ iff $x < 0$, and $\operatorname{sgn}(0) = 0$. Because of the following lemma (cp. [Gaßner 2009]), the minimum of the set in (6) exists.

Lemma 1. *For any polynomial $p \in \mathbb{R}[x] \setminus \mathbb{Q}[x]$, there is only a finite number of rational numbers $q \in \mathbb{Q}$ satisfying $p(q) \in \mathbb{Q}$.*

The Definition of $\mathcal{K}_{i,j}$ and the Constants $C_{i,1}, C_{i,2}, \ldots$. *For $i \geq 1$, let $N_{i,1}, N_{i,2}, \ldots$ be an enumeration of the set*

$$\{N_{\text{char}}(i, c_1, \ldots, c_{k_i}) \mid c_1, \ldots, c_{k_i} \in \mathbb{R}\}$$

such that $N_{i,j+1} > N_{i,j}$. For $i, j \geq 1$, let

$$\mathcal{K}_{i,j} = \{\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}} \mid \mathcal{B} \subseteq \mathbb{R}^\infty \;\&\; N_{i,j} = N_{\text{char}}(i, c_1, \ldots, c_{k_i})\}.$$

Moreover, let $C_{i,1} = N_{i,1}$ and, for $j \geq 2$, let $C_{i,j} = \max\{2^{C_{i,j-1}}, N_{i,j}\}$.
Since $C_{i,j} \geq N_{i,j} > \max\{\mu, \mu', \nu\}$, we have the following properties.

(i) By (5), $N_{i,j}$ is greater than any zero of the corresponding function f_k if $\text{degree}(f_k) \geq 1$. Therefore, by (4) we have

$$\nu_k = \operatorname{sgn}(f_k(N_{i,j})) = \operatorname{sgn}(f_k(C_{i,j})).$$

(ii) If an oracle machine $\mathcal{M} \in \mathcal{K}_{i,j}$ computes a positive integer N on input $\boldsymbol{x} \in \{0\}^{n_i-1} \times \{C_{i,j}\}$, then, by (6) there is a $k \leq s$ such that $\mu_k \neq 0$ and $N = f_k(C_{i,j})$. In this case, $f_k \in \mathbb{Q}[x]$ follows from (3). That means because of (2) that N is uniquely determined by μ_k and, consequently, by $N_{i,j}$.

(iii) A consequence of (7) is that $\mathcal{M} \in \mathcal{K}_{i,j}$ cannot compute the positive integers $C_{i,j+1}, C_{i,j+2}, \ldots$ on input $\boldsymbol{x} \in \{0\}^{n_i-1} \times \{C_{i,j}\}$ within $p_i(n_i)$ steps since these numbers are greater than $2^{C_{i,j}}$.

(iv) Property (5) implies also that $\mathcal{M} \in \mathcal{K}_{i,j}$ computes an integer $N \in \{0, 1\}$ on input $\boldsymbol{x} \in \{0\}^{n_i-1} \times \{C_{i,j}\}$ only if there is a $k \leq s$ such that $\text{degree}(f_k) = 0$ and consequently $f_k(x) = 0$ for all $x \in \mathbb{R}$ or $f_k(x) = 1$ for all $x \in \mathbb{R}$.

In the following construction, for any $\mathcal{B}$ and any $i, j \geq 1$, let $\mathcal{K}_{i,j}^{\mathcal{B}}$ be the subset of $\mathcal{K}_{i,j}$ given by

$$\mathcal{K}_{i,j}^{\mathcal{B}} = \{\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}} \mid N_{i,j} = N_{\mathrm{char}}(i, c_1, \ldots, c_{k_i})\}.$$

The Construction of $\mathcal{Q}_3$. *Let $m_0 = 0$. We construct the set $\mathcal{Q}_3$ in stages. Stage $i \geq 1$: Let n_i be an integer such that $n_i > m_{i-1}$, $p_i(n_i) < 2^{n_i-1}$, and $p_i(n_i) + n_i < 2^{n_i}$. Let $V_{i,0} = \emptyset$. Stage $j \geq 1$:*

$$W_{i,j} = \bigcup_{i'<i,} V_{i'} \cup \bigcup_{j'<j,} V_{i,j'},$$

$$V_{i,j} = \{\boldsymbol{x} \in \{0,1\}^{n_i-1} \times \{C_{i,j}\} \mid (\exists \mathcal{M} \in \mathcal{K}_{i,j}^{W_{i,j}})(\mathcal{M} \textit{ rejects } (0,\ldots,0,C_{i,j})$$

$$\& \ \boldsymbol{x} \textit{ is not queried by } \mathcal{M} \textit{ on input } (0,\ldots,0,C_{i,j}) \in \mathbb{N}^{n_i})\}.$$

Moreover, let $V_i = \bigcup_{j\geq 1} V_{i,j}$ and $m_i = 2^{n_i}$.
Finally, let $\mathcal{Q}_3 = \bigcup_{i\geq 1} V_i$ and

$$L_3 = \bigcup_{i\geq 1} \{(y_1,\ldots,y_{n_i-1},N) \in \{0,1\}^{n_i-1} \times \mathbb{N} \mid V_i \cap (\{0,1\}^{n_i-1} \times \{N\}) \neq \emptyset)\}.$$

The contents of the registers of any $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}}$ can be described by (1) if the input has the form $(0,\ldots,0,x) \in \mathbb{R}^{n_i}$. For any $\mathcal{B} \subseteq \mathbb{N}^\infty$, the value $N_{i,j}$ and the oracle $\mathcal{B}$ determine the computation path of any machine $\mathcal{N}_i^{\mathcal{B},c_1,\ldots,c_{k_i}} \in \mathcal{K}_{i,j}^{\mathcal{B}}$ traversed by the input $(0,\ldots,0,C_{i,j}) \in \mathbb{N}^{n_i}$ uniquely since (i) and (ii) hold. By (i), the result of a test of the form $f_k(C_{i,j}) \geq 0$ follows from $\mathrm{sgn}(v_k)$. The question $(f_{i_1}(C_{i,j}),\ldots,f_{i_t}(C_{i,j})) \in \mathcal{B}$? is answered no if one of the values $f_{i_1}(C_{i,j}),\ldots,f_{i_t}(C_{i,j})$ is not in $\mathbb{N}$. If the values are in $\mathbb{N}$, then, by (ii), the answer results from the values $\mu_{i_1},\ldots,\mu_{i_t}$ which are given by $N_{i,j}$.

Thus, the computation paths covered by $\mathcal{N}_i^{W_{i,j},c_1,\ldots,c_{k_i}} \in \mathcal{K}_{i,j}^{W_{i,j}}$ and by $\mathcal{N}_i^{\mathcal{Q}_3,c_1,\ldots,c_{k_i}} \in \mathcal{K}_{i,j}^{\mathcal{Q}_3}$
on $(0,\ldots,0,C_{i,j}) \in \mathbb{N}^{n_i}$ are even the same since we have also the following properties.

- By (iii), any $\boldsymbol{x} \in \{0,1\}^{n_i-1} \times \{C_{i,j+1}, C_{i,j+2}, \ldots\}$ is not queried.
- The length of the tuples in the oracle queries is less than 2^{n_i} and consequently less than n_{i+1} by definition of n_{i+1}.
- The machines $\mathcal{N}_i^{W_{i,j},c_1,\ldots,c_{k_i}}$ and $\mathcal{N}_i^{\mathcal{Q}_3,c_1,\ldots,c_{k_i}}$ do not query the tuples in $V_{i,j}$.

Moreover, for all $i, j \geq 1$, $p_i(n_i) < 2^{n_i-1}$ and (iv) imply that $V_{i,j}$ contains a tuple in $\{0,1\}^{n_i-1} \times \{C_{i,j}\}$ if a machine in $\mathcal{K}_{i,j}^{W_{i,j}}$ and, hence, any machine $\mathcal{N}_i^{W_{i,j},c_1,\ldots,c_{k_i}} \in \mathcal{K}_{i,j}^{W_{i,j}}$ and, consequently, any machine $\mathcal{N}_i^{\mathcal{Q}_3,c_1,\ldots,c_{k_i}} \in \mathcal{K}_{i,j}^{\mathcal{Q}_3}$ reject $(0,\ldots,0,C_{i,j}) \in \mathbb{R}^{n_i}$. That implies $L_3 \notin \mathrm{P}_{\mathbb{R}}^{\mathcal{Q}_3}$ and therefore the following.

Lemma 2. $L_3 \in \mathrm{DNP}_{\mathbb{R}}^{\mathcal{Q}_3} \setminus \mathrm{P}_{\mathbb{R}}^{\mathcal{Q}_3}$.

Proposition 4. *There is an oracle $\mathcal{Q}$ such that $\mathrm{P}_{\mathbb{R}}^{\mathcal{Q}} \neq \mathrm{DNP}_{\mathbb{R}}^{\mathcal{Q}}$.*

3 An Oracle Derived from the Knapsack Problem

The *Real Knapsack Problem*

$$\mathrm{KP}_{\mathbb{R}} = \bigcup_{n=1}^{\infty} \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid (\exists(\alpha_1, \ldots, \alpha_n) \in \{0,1\}^n)(\sum_{i=1}^{n} \alpha_i x_i = 1)\}$$

was introduced in [Blum et al. 1989] and studied, for instance, in [Koiran 1994] and [Meer 1992]. $\mathrm{KP}_{\mathbb{R}}$ belongs to $\mathrm{DNP}_{\mathbb{R}}$ since, for an input $(x_1, \ldots, x_n) \in \mathbb{R}^\infty$, a digital non-deterministic machine can guess any sequence $(\alpha_1, \ldots, \alpha_n) \in \{0,1\}^n$ and compute $\alpha_1 x_1 + \cdots + \alpha_n x_n$. It is not known whether $\mathrm{KP}_{\mathbb{R}} \in \mathrm{P}_{\mathbb{R}}$ holds.

Let $E_0 = \mathbb{Q}$, let $\tau_1, \tau_2, \ldots$ be a sequence of transcendental numbers such that τ_{i+1} is transcendental over $E_i =_{\mathrm{df}} E_{i-1}(\tau_i)$, and let the oracle $\mathcal{Q}_4$ and the decision problem L_4 be given.

$$A_n = \{(v_1, \ldots, v_{2n}) \in \{0, v\}^{2n} \mid v \in \mathbb{Z} \setminus \{0\} \ \&\ \textstyle\sum_{i=1}^{2n} v_i = nv\}.$$

$$\mathcal{Q}_4 = \textstyle\bigcup_{n=1}^{\infty} \{(\mathrm{sgn}(|v_1|), \ldots, \mathrm{sgn}(|v_{2n}|), \sum_{i=1}^{2n} v_i \tau_i) \in \mathbb{R}^{2n+1} \mid (v_1, \ldots, v_{2n}) \in A_n\}.$$

$$L_4 = \textstyle\bigcup_{n=1}^{\infty} \{(0, \ldots, 0, r) \in \mathbb{R}^{2n+1} \mid (\exists(v_1, \ldots, v_{2n}) \in A_n)(r = \sum_{i=1}^{2n} v_i \tau_i)\}.$$

Let us assume that the BSS machine $\mathcal{M}$ decides L_4 by using the oracle $\mathcal{Q}_4$ within a time bounded by a polynomial p and that $\mathcal{M}$ has only the constants $c_1, \ldots, c_k$. Let $F_0 = \bigcup_{i=0}^{\infty} E_i$. For $i = 1, \ldots, k$, let $F_i = F_{i-1}$ and $d_i = 1$ if $c_i \in F_{i-1}$, let $F_i = F_{i-1}(c_i)$ and $d_i = \infty$ if c_i is not algebraic over F_{i-1}, and let $F_i = F_{i-1}[c_i]$ if there is an irreducible polynomial $p_i \in F_{i-1}[x]$ of degree $d_i \geq 2$ with $p_i(c_i) = 0$. The value of any register computed by $\mathcal{M}$ on input $(0, \ldots, 0, x) \in \mathbb{R}^m$ can be described by some term of the form $\sum_{j_1, \ldots, j_k, j \leq 2^{p(m)}} \alpha_{j_1, \ldots, j_k, j} c_1^{j_1} \cdots c_k^{j_k} x^j$ where $\alpha_{j_1, \ldots, j_k, j} \in \mathbb{Z}$ and, consequently, by a polynomial of the form

$$q_j(x) = \tfrac{1}{r_0} \textstyle\sum_{j=0}^{2^{p(m)}} r_{j+1} x^j \text{ where}$$

$$r_j \quad = \sum_{\substack{m_1, \ldots, m_{i_0} \leq m_0 \\ j_s < \min\{d_s, j_0\}}} z_{m_1, \ldots, m_{i_0}, j_1, \ldots, j_k, j} \tau_1^{m_1} \cdots \tau_{i_0}^{m_{i_0}} c_1^{j_1} \cdots c_k^{j_k}$$

for some i_0, m_0, j_0, and $z_{m_1, \ldots, m_{i_0}, j_1, \ldots, j_k, j} \in \mathbb{Z}$ and $z_{m'_1, \ldots, m'_{i_0}, j'_1, \ldots, j'_k, 0} \neq 0$ for certain $m'_1, \ldots, m'_{i_0}, j'_1, \ldots, j'_k$. Thus, for the inputs of the form $(0, \ldots, 0, x) \in \mathbb{R}^m$, a non-trivial oracle query $(z_1, \ldots, z_s, q_j(x)) \in \mathcal{Q}_4$? (where $\mathrm{degree}(q_j) \geq 1$) can only be answered yes if $q_j(x) = \sum_{i=1}^{2n'} v'_i \tau_i$ is satisfied for some $(v'_1, \ldots, v'_{2n'}) \in A_{n'}$. Thus, we get the following.

Lemma 3. *Let $n > i_0$, $(0, \ldots, 0, v_{i_0+1}, \ldots, v_{2n}) \in A_n$, and $x = \sum_{i=i_0+1}^{2n} v_i \tau_i$. For $v_l \neq 0$, $v_{l+1} = \cdots = v_{2n} = 0$, a non-trivial oracle query $(z_1, \ldots, z_s, q_j(x)) \in \mathcal{Q}_4$? can be answered yes on inputs of the form $(0, \ldots, 0, x)$ only if $s \geq 2n$ and*

$$(z_{i_0+1}, \ldots, z_s) = (\mathrm{sgn}(|v_{i_0+1}|), \ldots, \mathrm{sgn}(|v_l|), 0, \ldots, 0).$$

Let n_0 be an even positive integer such that $n_0 > 2i_0$ and $p(2n_0+1) < 2^{\frac{n_0}{2}}$. Let P be the computation path of $\mathcal{M}$ described for inputs $(0,\dots,0,x)$ of size $2n_0+1$ uniquely by conditions of the form

$$(g_{j,1}(x),\dots,g_{j,s_j}(x)) \notin \mathcal{Q}_4 \quad (j \le t') \qquad \text{and} \qquad f_1(x) > 0,\dots,f_t(x) > 0$$

where $g_{j,1},\dots,g_{j,s_j}$ are polynomials, degree$(g_{j,s_j}) > 0$, and each f_j is defined by some equation of the form $f_j(x) = x^{n_j} + a_{n_j-1}x^{n_j-1} + \cdots + a_1x + a_0$.

Let $\tau > 0$ be transcendental over F_k and greater than all zeros of $f_1,\dots,f_t$. Then, $(0,\dots,0,\tau) \in \mathbb{R}^{2n_0+1} \setminus L_4$ traverses the path P. If $g_{j,1}(\tau),\dots,g_{j,s_j-1}(\tau) \in \{0,1\}$, then the polynomials $g_{j,1},\dots,g_{j,s_j-1}$ are constant. Since we have $|G| < 2^{\frac{n_0}{2}}$ for

$$G = \bigcup_{j<p(2n_0+1)} \{(g_{j,i_0+1}(x),\dots,g_{j,2n_0}(x)) \mid g_{j,i_0+1},\dots,g_{j,2n_0} \text{ are constant functions}\},$$

there is some $(0,\dots,0,x_0) \in \mathbb{R}^{2n_0+1}$ with $x_0 = \sum_{i=i_0+1}^{2n_0} w_i\tau_i$ satisfying

a) $(0,\dots,0,w_{i_0+1},\dots,w_{2n_0}) \in A_{n_0}$ and $w_{2n_0} \neq 0$,
b) $x_0 > \max(\{\tau\} \cup \bigcup_{\substack{j<p(2n_0+1)\\ s\le s_j-1}} \{x \mid g_{j,s}(x) \in \{0,1\} \ \&\ \text{degree}(g_{j,s}) \ge 1\})$,
c) $(\text{sgn}(|w_{i_0+1}|),\dots,\text{sgn}(|w_{2n_0}|)) \notin G$.

a) implies that $(0,\dots,0,x_0) \in L_4$. Moreover, we have $f_j(x_0) > 0$ by b). Therefore, by Lemma 3 and c), P is also traversed by $(0,\dots,0,x_0) \in \mathbb{R}^{2n_0+1}$. Hence, we get the following.

Lemma 4. $L_4 \in \text{DNP}_{\mathbb{R}}^{\mathcal{Q}_4} \setminus \text{P}_{\mathbb{R}}^{\mathcal{Q}_4}$.

Proposition 5. *There is an oracle $\mathcal{Q}$ which can be derived from* $\text{KP}_{\mathbb{R}}$ *such that* $\text{P}_{\mathbb{R}}^{\mathcal{Q}} \neq \text{DNP}_{\mathbb{R}}^{\mathcal{Q}}$.

References

[Baker et al. 1975] Baker, T., J. Gill, and R. Solovay: "Relativizations of the P =? NP question"; SIAM J. Comput. 4 (1975), 431–442.

[Blum et al. 1989] Blum, L., M. Shub, and S. Smale: "On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines"; Bulletin of the Amer. Math. Soc. 21 (1989), 1–46.

[Bürgisser 1999] Bürgisser, P.: "On the structure of Valiant's complexity classes"; Discrete Mathematics & Theoretical Computer Science 3(3) (1999), 73-94.

[Emerson 1994] Emerson, T.: "Relativizations of the P =? NP question over the reals (and other ordered rings)"; Theoretical Computer Science 133 (1994), 15–22.

[Gaßner 2008] Gaßner, C.: "On the power of relativized versions of P, DNP, and NP for groups"; (2008) Submitted.

[Gaßner 2009] Gaßner, C.: "Oracles and relativizations of the P =? NP question for several structures"; JUCS vol. 15 no. 6 (2009), 1186–1205.

[Koiran 1994] Koiran, P.: "Computing over the reals with addition and order"; Theoretical Computer Science 133 (1994), 35–47.

[Meer 1992] Meer, K.: "A note on a P ≠ NP result for a restricted class of real machines"; Journal of Complexity 8 (1992), 451–453.

[Poizat 1995] Poizat, B.: "Les Petits Cailloux"; Aléas (1995).