On Relativizations of the $P \not= NP$ Question for Several Structures

Christine Gaßner
Greifswald

Hagen 2008
On Relativizations of the $P \nleq \text{NP}$ Question for Several Structures

Our goal:

- **Construct several oracles with**
  \[ P^A \neq \text{NP}^A \]
  with respect to the uniform model of computation.

- **Evaluate known constructions**
  using knowledge
  - of the mathematical logic,
  - about the ring over the real numbers.

- **Show difficulties in deriving a structure with**
  \[ P = \text{NP} \]
  from an oracle with \( P^A = \text{NP}^A \).
On Relativizations of the P $\not\equiv$ NP Question for Several Structures

1. The uniform model of computation
2. Diagonalization techniques and halting problems
3. Structures and oracles with $P^A \neq NP^A$
4. Structures and an oracle with $P^A = NP^A$
The uniform model of computation

A structure: \( \Sigma = (U; c_1, \ldots, c_u; f_1, \ldots, f_v; R_1, \ldots, R_w, =) \)
\[ \Sigma = (U; (c_i)_{i \in F}; (f_i)_{i \in G}; (R_i)_{i \in H}, =) \]

Computation:
\[
\begin{align*}
&l: Z_k := f_j(Z_{k1}, \ldots, Z_{km_j}); \\
&l: Z_k := c_j;
\end{align*}
\]

Branching:
\[
\begin{align*}
&l: \text{if } R_j(Z_{k1}, \ldots, Z_{kn_j}) \text{ then goto } l_1 \text{ else goto } l_2; \\
&l: \text{if } Z_k = Z_j \text{ then goto } l_1 \text{ else goto } l_2;
\end{align*}
\]

Copy:
\[
\begin{align*}
&l: Z_{Ik} := Z_{Ij};
\end{align*}
\]

Index computation:
\[
I_k := 1; \quad I_k := I_k + 1; \quad \text{if } I_k = I_j \text{ then goto } l_1 \text{ else goto } l_2;
\]
Examples for several structures

\[ \mathbb{Z}_2 = (\{0, 1\}; 0, 1; +, \cdot; =) \quad (\Rightarrow \text{Turing machines}) \]

\[ \mathbb{R} = (\mathbb{R}; \mathbb{R}; +, -, \cdot; \leq) \quad (\Rightarrow \text{BSS model}) \]

\[ \Sigma_{\text{string}} = (\{0, 1\}^*; \varepsilon, 0, 1; \text{add}, \text{sub}_1, \text{sub}_r; =) \]

\[ \Sigma_{\text{tree}} = (\text{tree(}\mathbb{R}\text{)}; \text{nil}; \text{concat}, \text{root}, \text{sub}_1, \text{sub}_r; =) \]

\[ \text{concat}(a, t_1, t_2) = \begin{array}{c} a \\ t_1 \\ t_2 \end{array} \]
The input: \((Z_1, \ldots, Z_n) := (x_1, \ldots, x_n); I_1 := n; I_2 := 1; \ldots; I_{k_M} := 1;\)
Computation in polynomial time

For any machine $M$ there is some polynomial $p_M$ such that

$M$ halts for $x = (x_1, \ldots, x_n)$ within $p_M(n)$ steps.

One operation is executed within one time unit.

$\Rightarrow P_\Sigma \subseteq \text{DEC}_\Sigma$  \quad (P_\Sigma \triangleq \text{problems are decidable in polynomial time})
Why do we consider the uniform model of computation?

In describing algorithms (for instance, in the computational geometry) we often use
- models over algebraic structures with several costs for operations,
- the BSS model with unit cost measure.

Important:
- to investigate
  - common properties
  - differences
- of several models, in order to answer:
  - When can we use a model over an algebraic structure?
  - Which simplification can imply problems?
  - Which properties are necessary in order to get a special complexity for a problem?
The non-deterministic instructions

The non-determinism:

\[ \text{guess}(Z_k); \quad \text{Arbitrary elements can be guessed!} \]

\[ \Rightarrow \quad P_{\Sigma} \subseteq \text{NP}_{\Sigma} \]
Some $P_{\Sigma} \not\equiv NP_{\Sigma}$ problems for several structures

<table>
<thead>
<tr>
<th>$\Sigma$</th>
<th>$P_{\Sigma} = NP_{\Sigma}$?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\mathbb{C}; \mathbb{C}; +, -, \cdot; =)$</td>
<td>?</td>
</tr>
<tr>
<td>$(\mathbb{R}; \mathbb{R}; +, -; \leq)$</td>
<td>?</td>
</tr>
<tr>
<td>$(\mathbb{R}; \mathbb{R}; +, -; =)$</td>
<td>no $(\leq)$</td>
</tr>
<tr>
<td>$(\mathbb{R}; \mathbb{R}; +, -; \leq)$</td>
<td>?</td>
</tr>
<tr>
<td>$(\mathbb{R}; \mathbb{R}; +, -; =)$</td>
<td>no (Meer / Koiran)</td>
</tr>
<tr>
<td>$(\mathbb{Z}; \mathbb{Z}; +, -; \leq)$</td>
<td>no (even integers)</td>
</tr>
<tr>
<td>$(\mathbb{Z}; \mathbb{Z}; +, -; =)$</td>
<td>no (even integers)</td>
</tr>
<tr>
<td>$(\mathbb{Z}; 1; (\phi_s)_{s \in \mathbb{Z}}; =) \quad \phi_s(x) = sx$</td>
<td>no (no NP-complete problem)</td>
</tr>
</tbody>
</table>
Halting problems for $\Sigma$

\[ H_\Sigma = \{ (x_1, \ldots, x_n, \text{Code}(M)) \mid x \in U^* \& M \text{ is a deterministic } \Sigma\text{-machine} \]

\[ H_\Sigma^{\text{spec}} = \{ \text{Code}(M) \mid M \text{ is a deterministic } \Sigma\text{-machine} \]

\[ \& M \text{ halts on } \text{Code}(M) \} \]
## Diagonalization techniques

The undecidability of the Halting problem $H_\Sigma$ (for Turing machines)

1. **The set of machines is countable.** Assume that $H_\Sigma^{\text{spec}}$ is decidable.

<table>
<thead>
<tr>
<th>Halt?</th>
<th>bin(1)</th>
<th>...</th>
<th>bin(i)</th>
<th>...</th>
<th>bin(j)</th>
<th>...</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>yes / no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_i$</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_j$</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_j$</td>
<td>no / yes</td>
<td>...</td>
<td>no</td>
<td>...</td>
<td>yes</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

⇒ **There is an** $M$ **recognizing the complement of** $H_\Sigma^{\text{spec}}$.    

CCA 2008  
gassnerc@uni-greifswald.de
2. The codes of machines are ordered. Assume that $H_\Sigma^{\text{spec}}$ is decidable.

<table>
<thead>
<tr>
<th>Halt?</th>
<th>...</th>
<th>...</th>
<th>Code($M_i$)</th>
<th>...</th>
<th>Code($M_j$)</th>
<th>...</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>:</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_i$</td>
<td></td>
<td></td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_j$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M$</td>
<td></td>
<td></td>
<td>no</td>
<td></td>
<td>yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\Rightarrow$ There is an $M$ recognizing the complement of $H_\Sigma^{\text{spec}}$. ☢️
Diagonalization techniques
The undecidability of the Halting problem $H_\Sigma$ (for any structure)

3. $\Sigma$ arbitrary (We can generalize the result.)

Assume: $H_\Sigma$ is decidable.

$\Rightarrow$ $H_\Sigma^{\text{spec}}$ is decidable.

$\Rightarrow$ The complement of $H_\Sigma^{\text{spec}}$ is semi-decidable by a $\Sigma$-machine $M$.

$\Rightarrow M$ halts on $\text{Code}(M)$

$\iff M$ does not halt on $\text{Code}(M)$.

$\Rightarrow$ $\not\exists$
Oracle machines

**Oracle query:**

\[ l : \text{ if } (Z_1, \ldots, Z_{l_1}) \in B \text{ then goto } l_1 \text{ else goto } l_2 ; \]

The length can be computed by \( I_1 := 1; \ I_i := I_{i-1} + 1; \ldots \)

**B oracle,** \( B \subseteq U^\infty = \bigcup_{n \geq 1} U^n \)

We will define oracles such that

\[ P_\Sigma^Q \neq NP_\Sigma^Q, \]

\[ P_\Sigma^O = NP_\Sigma^O. \]

(cp. also Baker, Gill, and Solovay; Emerson; ... for Turing machines... )
An oracle $Q$ with $P^Q_\Sigma \neq NP^Q_\Sigma$

Diagonalization techniques by Baker, Gill, and Solovay

1. If the set of programs is countable, for any oracle $B \subseteq U^\infty$,

let $N_i^B$ be the $P^B_\Sigma$-machine
  - executing $p_i(n)$ instructions of program $P_i$ for any $x \in U^n$. $a, b \in U$.

Proposition: $\{y \mid (\exists \ i \geq 1)(y \in U^{n_i} \land V_i \neq \emptyset)\} \in NP^Q_\Sigma \setminus P^Q_\Sigma$. 

CCA 2008
gassnerc@uni-greifswald.de
### An oracle $Q$ with $P^Q \Sigma \neq NP^Q \Sigma$

Diagonalization techniques by Baker, Gill, and Solovay

The set of programs is countable. $a, b \in U$.

<table>
<thead>
<tr>
<th>$P_i$</th>
<th>$p_i$</th>
<th>$n_i$</th>
<th>Length in a query</th>
<th>$(a, \ldots, a) \in U^{n_i}$</th>
<th>$Q = Q_\Sigma = \bigcup_{i \geq 1} W_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$</td>
<td>$p_1$</td>
<td>$p_1(n_1) + n_1 &lt; 2^{n_1}$</td>
<td>$\leq p_1(n_1) + n_1 &lt; n_2$</td>
<td>rejected</td>
<td>$W_1 = { x</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td></td>
<td>accepted</td>
<td></td>
<td>$W_1 = \emptyset$</td>
</tr>
<tr>
<td>$P_i$</td>
<td>$p_i$</td>
<td>$2^{n_i-1} &lt; n_i$</td>
<td>$p_i(n_i) + n_i &lt; 2^{n_i}$</td>
<td>rejected</td>
<td>$W_{i+1} = W_i \cup { x</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td></td>
<td>accepted</td>
<td></td>
<td>$W_{j+1} = W_j$</td>
</tr>
</tbody>
</table>

$\Rightarrow N_u^{W_i}$ rejects $(a, \ldots, a) \iff N_u^{W_{i+1}}$ rejects $(a, \ldots, a) \iff N_u^Q$ rejects $(a, \ldots, a)$. 

CCA 2008
gassnerc@uni-greifswald.de
An oracle $Q$ with $P_{\Sigma}^Q \neq NP_{\Sigma}^Q$

Diagonalization techniques by Baker, Gill, and Solovay

1. If the set of programs is **countable**, for any oracle $B \subseteq U^\infty$,

   let $N_i^B$ be the $P_{\Sigma}^B$-machine
   - executing $p_i(n)$ instructions of program $P_i$ for any $x \in U^n$. $a, b \in U$.

   $V_0 = \emptyset$, $m_0 = 0$.
   Stage $i \geq 1$: Let $n_i > m_{i-1}$, $m_i = 2^{n_i}$, $p_i(n_i) + n_i < m_i$.
   $W_i = \bigcup_{j < i} V_j$

   $V_i = \{x \in U^{n_i} \mid N_i^{W_i}$ rejects $(a, \ldots, a) \in U^{n_i}$
   
   & $x$ is not queried by $N_i^{W_i}$ on $(a, \ldots, a) \in U^{n_i}\}$

   $Q = Q_\Sigma = \bigcup_{i \geq 1} W_i$

**Proposition:** $\{y \mid (\exists i \geq 1)(y \in U^{n_i} \& V_i \neq \emptyset)\} \in NP_{\Sigma}^Q \setminus P_{\Sigma}^Q$. 
2. If $U$ is ordered, for suitable codes $u \in U \subseteq U^\infty$ and any oracle $B \subseteq U^\infty$, let $N_u^B$ be the $P_\Sigma^B$-machine

- executing $p_u(n)$ instructions of program $P_u$ for any $x \in U^n$.

$N \subseteq U$.

Proposition: $\{ y | (\exists \ n \geq 2) \ ((n, y) \in Q_\Sigma) \} \in \text{NP}_\Sigma^Q \setminus \text{P}_\Sigma^Q$. 

Diagonalization techniques by Emerson
An oracle $Q$ with $P_{\Sigma}^Q \neq NP_{\Sigma}^Q$

Diagonalization techniques by Emerson

$U$ ordered and $\mathbb{N} \subseteq U$.

| $K_i$ | elements in a query on $u \in K_i$ within a time period bounded by $p_u(|u|)$ | $Q = Q_\Sigma = \cup_{i \geq 1} W_i$ |
|------|---------------------------------------------------------------------------|----------------------------------|
| $K_1$ | $\leq 1$                                                                   | $W_1 = \emptyset$                |
| $\vdots$ | $\leq i$                                                                   | $W_{i+1} = W_i \cup \{ (i + 1, u) \mid u \in K_i \text{ and } N_u^{W_i} \text{ rejects } u \}$ |

$\Rightarrow N_u^{W_i} \text{ rejects } u \iff N_u^Q \text{ rejects } u.$

CCA 2008
gassnerc@uni-greifswald.de
An oracle $Q$ with $\mathsf{P}_\Sigma^Q \neq \mathsf{NP}_\Sigma^Q$

Diagonalization techniques by Emerson

2. If $U$ is ordered, for suitable codes $u \in U \subseteq U^\infty$ and any oracle $B \subseteq U^\infty$,

let $N_u^B$ be the $\mathsf{P}_\Sigma^B$-machine

- executing $p_u(n)$ instructions of program $P_u$ for any $x \in U^n$.

$N \subseteq U$.

---

$V_0 = \emptyset$.

Stage $i \geq 1$:

$K_i = \{ u \in U \mid (\forall j > i)(\forall B \subseteq U^\infty) (j \in U \text{ is not queried by } N_u^B \text{ on } u) \}$

$W_{i+1} = W_i \cup \{ (i+1, u) \mid u \in K_i \& N_u^{W_i} \text{ rejects } u \}$

$Q = Q_\Sigma = \bigcup_{i \geq 1} W_i$

**Proposition:** $\{ y \mid (\exists n \geq 2) ((n, y) \in Q_\Sigma) \} \in \mathsf{NP}_\Sigma^Q \setminus \mathsf{P}_\Sigma^Q$. 
An oracle $Q$ with $P^Q_\Sigma \neq NP^Q_\Sigma$

Diagonalization techniques (a generalization)

3. If $U$ is infinite, for suitable codes $u \in U \subseteq U^\infty$ and any oracle $B \subseteq U^\infty$, let $N^B_u$ be the $P^B_\Sigma$-machine

- executing $p_u(n)$ instructions of program $P_u$ for any $x \in U^n$.

$\alpha_1, \alpha_2, \alpha_3, \ldots \in U$.

Proposition: $\{y \mid (\exists n \geq 2) (\langle u_{\alpha_2}, y \rangle \in Q^y)\} \in NP^Q_\Sigma \setminus P^Q_\Sigma$. 

CCA 2008
gassnerc@uni-greifswald.de
**An oracle \( Q \) with \( P_{\Sigma}^Q \neq NP_{\Sigma}^Q \)**

Diagonalization techniques (a generalization)

\( \Sigma \) arbitrary, \( \alpha_1, \alpha_2, \alpha_3, \ldots \in U. \)

| \( K_i \) | elements in a query on \( u \in K_i \) within a time period bounded by \( p_u(|u|) \) |
|---|---|
| \( K_1 \) | \( \notin \{\alpha_1, \alpha_2, \alpha_3, \ldots\} \) |
| \( \vdots \) | \( W_1 = \emptyset \) |
| \( K_i \) | \( \notin \{\alpha_{i+1}, \alpha_{i+2}, \alpha_{i+3}, \ldots\} \) |
| \( \vdots \) | \( W_{i+1} = W_i \cup \{ (\alpha_{i+1}, u) \mid u \in K_i \text{ and } N_u^{W_i} \text{ rejects } u \} \) |

\[ \Rightarrow N_u^{W_i} \text{ rejects } u \iff N_u^{Q} \text{ rejects } u. \]
An oracle $Q$ with $P^Q_\Sigma \neq NP^Q_\Sigma$

Diagonalization techniques (a generalization)

3. If $U$ is infinite, for suitable codes $u \in U \subseteq U^\infty$ and any oracle $B \subseteq U^\infty$, let $N^B_u$ be the $P^B_\Sigma$-machine

- executing $p_u(n)$ instructions of program $P_u$ for any $x \in U^n$.

$\alpha_1, \alpha_2, \alpha_3, \ldots \in U$.

$V_0 = \emptyset$.

Stage $i \geq 1$:

$K_i = \{u \in U \mid (\forall j > i)(\forall B \subseteq U^\infty)$

$(N^B_u$ does not compute or use the value $\alpha_j$ on $u))\}$

$W_{i+1} = W_i \cup \{(\alpha_{i+1}, u) \mid u \in K_i \& N^W_u$ rejects $u\}$

$Q = Q_\Sigma = \bigcup_{i \geq 1} W_i$

**Proposition:** $\{y \mid (\exists \ n \geq 2) \ ((\alpha_n, y) \in Q_\Sigma)\} \in NP^Q_\Sigma \setminus P^Q_\Sigma$. 
An oracle $Q$ with $P^Q_\Sigma \neq NP^Q_\Sigma$

Using the undecidability of the Halting problem $H_\Sigma$

4. $U$ infinite,
a finite number of operations and relations,

$$\{a_1, a_2, a_3, \ldots\} \subseteq U$$ enumerable and decidable.

$$Q = Q_\Sigma = \{ (a_t, x, \text{Code}(M)) \mid$$

$$x \in U^\infty \& M \text{ is a deterministic } \Sigma\text{-machine}$$

$$\& \ M(x) \downarrow^t \}$$

$M$ accepts $x = (x_1, \ldots, x_n) \in U^\infty$ within $t$ steps.

**Proposition:** $H_\Sigma \in NP^Q_\Sigma \setminus P^Q_\Sigma$. \quad (P^Q_\Sigma \subseteq \text{DEC}_\Sigma)$
An oracle $O_\Sigma$ with $P_\Sigma^{O_\Sigma} = NP_\Sigma^{O_\Sigma}$

A universal oracle:

$$\in U^U$$

$$O = O_\Sigma = \{ (b, \ldots, b, x, \text{Code}(M)) \mid x \in U^\infty \& M \text{ is a non-deterministic } \Sigma\text{-machine using } O$$

$$\& M(x) \downarrow \}$$

**Proposition:** $P_\Sigma^{O} = NP_\Sigma^{O}$. 

CCA 2008  
gassnerc@uni-greifswald.de
An oracle $O_{\Sigma}$ containing only tuples of length 1 with $P_{\Sigma}^{O_{\Sigma}} = NP_{\Sigma}^{O_{\Sigma}}$?

Structures over strings

$\Sigma = (U^*; \varepsilon, a, b, c_3, \ldots, c_u; \text{add}, \text{sub}_l, \text{sub}_r, f_1, \ldots, f_v; R_1, \ldots, R_w, =)$

$$(d_1, \ldots, d_k) \in U^k \subset U^\infty \quad \text{stored in } k \text{ registers}$$

$$s = d_1 \cdots d_k \in U^* \quad \text{stored in one register}$$

$$d \in U$$

$${\text{add}}(s, d) = sd \quad {\text{sub}}_l(sd) = s \quad {\text{sub}}_r(sd) = d$$

CCA 2008
gassnerc@uni-greifswald.de
An oracle $O_\Sigma$ containing only tuples of length 1 with $P_\Sigma^{O_\Sigma} = NP_\Sigma^{O_\Sigma} \ ?$

Recall: $P_\Sigma^{O_\Sigma} = NP_\Sigma^{O_\Sigma}$ and $P_\Sigma^{Q_\Sigma} \neq NP_\Sigma^{Q_\Sigma}$ for

$$O_\Sigma = \{ (b_1, \ldots, b_t, x, \text{Code}(M)) \mid x \in (U^*)^\infty$$

$t \times$

& $M$ is a non-deterministic $\Sigma$-machine using $O_\Sigma$ & $M(x) \downarrow$

$$Q_\Sigma = \{ (b_1 \cdots b_t, x, \text{Code}(M)) \mid x \in (U^*)^\infty$$

$t \times$

& $M$ is a deterministic $\Sigma$-machine & $M(x) \downarrow$

**Theorem:** There is not an oracle $O$ with $b_1 \cdots b_t \cdot \text{string}(x) \cdot \text{string}(	ext{Code}(M)) \in O$

$\iff x \in (U^*)^\infty$

& $M$ is a non-deterministic $\Sigma$-machine using $O$

& $M(x) \downarrow$. 

No set!
An additional relation $R$ on padded codes of the members of a universal oracle $O$ with $P_{\Sigma^O} = NP_{\Sigma^O}$

Binary trees with decidable identity relation (Gaßner, Dagstuhl 2004)

Strings with operations for adding and deleting the last character (Gaßner, CiE 2007)
Z as oracle with $P^Z_R \neq \text{NP}^Z_R$

Using the properties of $(\mathbb{R}; \mathbb{R}; +, -, \cdot; \leq)$

5. $\Sigma = (\mathbb{R}; \mathbb{R}; +, -, \cdot; \leq)$ or $\Sigma = (\mathbb{R}; \mathbb{R}; +, -, \cdot; =)$

$Q \in \text{NP}^Z_R$.

Program: guess($y_1$); guess($y_2$); if $y_1, y_2 \in \mathbb{Z}$, $y_1 \neq 0$ and $y_1 x = y_2$ then output 1.

Proposition: $P^Z_R \neq \text{NP}^Z_R$. 
\[\mathbb{Z} \text{ as oracle with } P^\mathbb{Z}_R \neq NP^\mathbb{Z}_R\]

Using the properties of \((\mathbb{R}; \mathbb{R}; +, -, \cdot; \leq)\)

5. \(\Sigma = (\mathbb{R}; \mathbb{R}; +, -, \cdot; \leq)\) or \(\Sigma = (\mathbb{R}; \mathbb{R}; +, -, \cdot; =)\)

\[
\mathbb{Q} \in NP^\mathbb{Z}_R.
\]

Program: \text{guess}(y_1); \text{guess}(y_2); \text{if } y_1, y_2 \in \mathbb{Z}, y_1 \neq 0 \text{ and } y_1x = y_2 \text{ then output } 1.

Assume that \(\mathbb{Q}\) is decidable by a machine \(M\).

Description of any computation path by a system of conditions of the form

\[
p_k(x) \in \mathbb{Z} \quad p_k(x) \notin \mathbb{Z} \quad p_k(x) \leq 0 \quad p_k(x) < 0 \quad (k \leq m).
\]

\(\Rightarrow\) There are \(r \notin \mathbb{Q} \cup \{x \mid p_k(x) \in \mathbb{Z}\}\) and \((q_i)_i \in \mathbb{N}\) such that \(q_i \in \mathbb{Q}\) and \(q_i \rightarrow r.\)

\(\Rightarrow\) \(r\) and some \(q_j\) satisfy the same conditions \(p_k(x) \notin \mathbb{Z}\) and \(p_k(x) < 0.\)

\(\Rightarrow\) \(r\) and \(q_j\) are rejected. \(\Rightarrow\) \(\nexists\)

Proposition: \(P^\mathbb{Z}_R \neq NP^\mathbb{Z}_R.\)
On Relativizations of the $P \not\equiv NP$ Question for Several Structures

Thank you for your attention!

Christine Gaßner
Greifswald.

Thanks also to
Robert Bialowons,
Volkmar Liebscher,
Rainer Schimming.