On Relativizations of the $P \stackrel{?}{=} NP$ Question for Several Structures

Christine Gaßner
Greifswald

Hagen 2008

On Relativizations of the $P \stackrel{?}{=} NP$ Question for Several Structures

Our goal:

- Construct several oracles with
  $$P^A \neq NP^A$$
  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
  $$P = NP$$
  from an oracle with $P^A = NP^A$.

gassnerc@uni-greifswald.de

On Relativizations of the $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ Question for Several Structures

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

gassnerc@uni-greifswald.de

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: $l\colon\ Z_k := f_j(Z_{k_1},\ldots,Z_{k_{m_j}});$

$l\colon\ Z_k := c_j;$

Branching: $l\colon$ if $R_j(Z_{k_1},\ldots,Z_{k_{n_j}})$ then goto l_1 else goto l_2;

$l\colon$ if $Z_k = Z_j$ then goto l_1 else goto l_2;

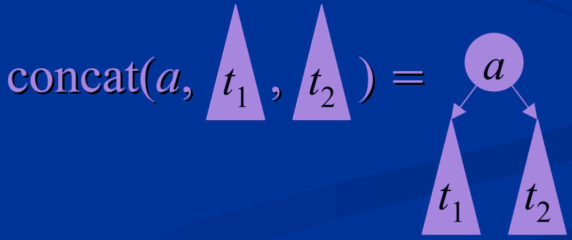
Copy: $l\colon\ Z_{I_k} := Z_{I_j};$

Index computation: $I_k := 1;\quad I_k := I_k + 1;\quad$ if $I_k = I_j$ then goto l_1 else goto l_2;

gassnerc@uni-greifswald.de

Examples for several structures

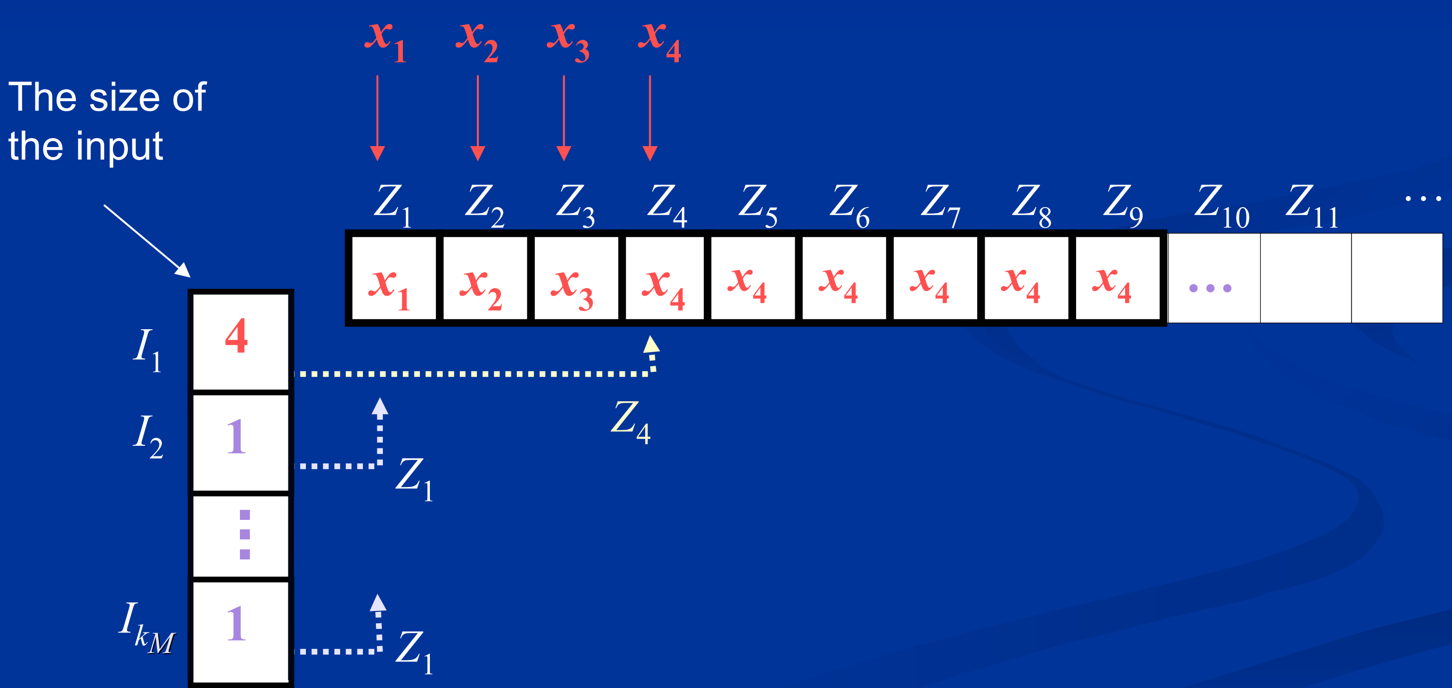
$\mathbb{Z}_2$	$= (\{0, 1\}; 0, 1; +, \cdot\ ; =)$	(⇨ Turing machines)
$\mathbb{R}$	$= (\mathbb{R}; \mathbb{R}; +, -, \cdot\ ; \leq)$	(⇨ BSS model)
Σ_{string}	$= (\{0, 1\}^*; \varepsilon, 0, 1; \text{add}, \text{sub}_l, \text{sub}_r; =)$	
Σ_{tree}	$= (\text{tree}(\mathbb{R}); \text{nil}; \text{concat}, \text{root}, \text{sub}_l, \text{sub}_r; =)$	



gassnerc@uni-greifswald.de

The machine and the input

The input: $(Z_1,\ldots,Z_n) := (x_1,\ldots,x_n);\ I_1 := n;\ I_2 := 1;\ldots;\ I_{k_M} := 1;$



gassnerc@uni-greifswald.de

Computation in polynomial time

For any machine M there is some polynomial p_M such that

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

One operation is executed within one time unit.

$\Rightarrow \mathrm{P}_\Sigma \subseteq \mathrm{DEC}_\Sigma$ ($\mathrm{P}_\Sigma \triangleq$ problems are decidable in polynomial time)

gassnerc@uni-greifswald.de

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?

gassnerc@uni-greifswald.de

The non-deterministic instructions

The non-deterministic:

$\text{guess}(Z_k);$ Arbitrary elements can be guessed!

⇨ $P_\Sigma \subseteq NP_\Sigma$

gassnerc@uni-greifswald.de

Some $P_\Sigma \stackrel{?}{=} NP_\Sigma$ problems for several structures

Σ	$P_\Sigma = NP_\Sigma$?
$(\mathbb{C}\,;\, \mathbb{C}\,;\, +, -, \cdot\,;\; =)$	**?**
$(\mathbb{R}\,;\, \mathbb{R}\,;\, +, -, \cdot\,;\; \leq)$	**?**
$(\mathbb{R}\,;\, \mathbb{R}\,;\, +, -, \cdot\,;\; =)$	no ($\leq$)
$(\mathbb{R}\,;\, \mathbb{R}\,;\, +, -\,;\; \leq)$	**?**
$(\mathbb{R}\,;\, \mathbb{R}\,;\, +, -\,;\; =)$	no (Meer / Koiran)
$(\mathbb{Z}\,;\, \mathbb{Z}\,;\, +, -\,;\, \leq)$	no (even integers)
$(\mathbb{Z}\,;\, \mathbb{Z}\,;\, +, -\,;\, =)$	no (even integers)
$(\mathbb{Z}\,;\, 1;\, (\varphi_s)_{s\in\mathbb{Z}};\, =)$ $\quad \varphi_s(x) = sx$	no (no NP-complete problem)

gassnerc@uni-greifswald.de

Halting problems for Σ

$$H_\Sigma = \{(\overbrace{x_1,\ldots,x_n}^{\boldsymbol{x}}, Code(M)) \mid \boldsymbol{x} \in U^\infty \;\&\; M \text{ is a deterministic } \Sigma\text{-machine} \;\&\; M \text{ halts on } \boldsymbol{x}\}$$

$$H_\Sigma^{\text{spec}} = \{Code(M) \mid M \text{ is a deterministic } \Sigma\text{-machine} \;\&\; M \text{ halts on } Code(M)\}$$

gassnerc@uni-greifswald.de

Diagonalization techniques

The undecidability of the Halting problem H_Σ (for Turing machines)

1. The set of machines is countable. Assume that H_Σ^{spec} is decidable.

Halt?	bin(1)	⋯	bin(*i*)	⋯	bin(*j*)	⋯	⋯
M_1	yes / no						
⋮		⋯					
M_i			yes				
⋮				⋯			
M_j					no		
⋮						⋯	
⋮							
M	no / yes	⋯	no	⋯	yes	⋯	⋯

⇒ There is an M recognizing the complement of H_Σ^{spec}. ↯

Diagonalization techniques

The undecidability of the Halting problem H_Σ (for $(\mathbb{R}\,;\,\mathbb{R}\,;\,+,-,\cdot\,;\,\leq)$)

2. The codes of machines are ordered. Assume that H_Σ^{spec} is decidable.

Halt?	⋯	⋯	$Code(M_i)$	⋯	$Code(M_j)$	⋯	⋯
⋮	⋯						
⋮		⋯					
M_i			yes				
⋮				⋯			
M_j					no		
⋮						⋯	
⋮							
M	⋯	⋯	no	⋯	yes	⋯	⋯

⇒ There is an M recognizing the complement of H_Σ^{spec}. ↯

gassnerc@uni-greifswald.de

Diagonalization techniques

The undecidability of the Halting problem H_Σ (for any structure)

3. Σ arbitrary (We can generalize the result.)

Assume: H_Σ is decidable.

$\Rightarrow$ H_Σ^{spec} is decidable.

$\Rightarrow$ The complement of H_Σ^{spec} is semi-decidable by a Σ-machine M.

$\Rightarrow$ M halts on $Code(M)$

$\Leftrightarrow$ M does not halt on $Code(M)$.

$\Rightarrow$ ↯

gassnerc@uni-greifswald.de

Oracle machines

Oracle query:

$$l\text{: if } (\underbrace{Z_1,\ldots,Z_{I_1}}) \in B \text{ then goto } l_1 \text{ else goto } l_2\,;$$

The length can be computed by I_1 := 1; I_1 := I_1 + 1; ....

B oracle, $B \subseteq U^\infty = \cup_{n\geq 1} U^n$

We will define oracles such that

$$\mathrm{P}_\Sigma^{\;Q} \neq \mathrm{NP}_\Sigma^{\;Q},$$

$$\mathrm{P}_\Sigma^{\;O} = \mathrm{NP}_\Sigma^{\;O}.$$

(cp. also Baker, Gill, and Solovay; Emerson; ... for Turing machines...)

gassnerc@uni-greifswald.de

An oracle Q with $\mathrm{P}_\Sigma^{Q} \neq \mathrm{NP}_\Sigma^{Q}$

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 $\mathrm{P}_\Sigma{}^B$-machine

- executing $p_i(n)$ instructions of program P_i for any $x \in U^n$. $a, b \in U$.

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

gassnerc@uni-greifswald.de

An oracle Q with $P_\Sigma^Q \neq NP_\Sigma^Q$

Diagonalization techniques by Baker, Gill, and Solovay

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

P_i	p_i	n_i	Length in a query	$(a,\ldots,a) \in U^{n_i}$	$Q = Q_\Sigma = \cup_{i \geq 1} W_i$
P_1	p_1	$p_1(n_1) + n_1 < 2^{n_1}$	$\leq p_1(n_1) + n_1 < n_2$	rejected	$W_1 = \{ \boldsymbol{x} \mid \boldsymbol{x} \in U^{n_1}$ & $\boldsymbol{x}$ not queried...$\}$
				accepted	$W_1 = \varnothing$
⋮	⋮				
P_i	p_i	$2^{n_i - 1} < n_i$ $p_i(n_i) + n_i < 2^{n_i}$	$\leq p_i(n_i) + n_i$ $< n_{i+1}$	rejected	$W_{i+1} = W_i \cup \{ \boldsymbol{x} \mid \boldsymbol{x} \in U^{n_i}$ & $\boldsymbol{x}$ not queried by $N_i^{W_i}$ on $(a,\ldots,a) \in U^{n_i}\}$
⋮	⋮				
P_j	p_j	$2^{n_j - 1} < n_j$ $p_j(n_j) + n_j < 2^{n_j}$	$\leq p_j(n_j) + n_j$ $< n_{j+1}$	accepted	$W_{j+1} = W_j$
⋮	⋮			...	

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

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 $\boldsymbol{x} \in U^n$. $a, b \in U$.

$V_0 = \emptyset$, $m_0 = 0$.

Stage $i \geq 1$: Let $n_i > m_{i-1}$, $m_i = 2^{n_i}$, $p_i(n_i) + n_i < m_i$.

$W_i = \cup_{j<i} V_j$

$V_i = \{\boldsymbol{x} \in U^{n_i} \mid N_i^{W_i}$ rejects $(a,\ldots, a) \in U^{n_i}$

$\quad\quad \&\ \boldsymbol{x}$ is not queried by $N_i^{W_i}$ on $(a,\ldots, a) \in U^{n_i}\}$

Diagonalization technique

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

Proposition: $\{\boldsymbol{y} \mid (\exists\ i \geq 1)(\ \boldsymbol{y} \in U^{\,n_i}\ \ \&\ \ V_i \neq \emptyset)\} \in NP_\Sigma^{\,Q} \setminus P_\Sigma^{\,Q}$.

gassnerc@uni-greifswald.de

An oracle Q with $P_\Sigma{}^Q \neq NP_\Sigma{}^Q$

Diagonalization techniques by Emerson

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

let $N_{\boldsymbol{u}}{}^B$ be the $P_\Sigma{}^B$-machine

- executing $p_{\boldsymbol{u}}(n)$ instructions of program $P_{\boldsymbol{u}}$ for any $\boldsymbol{x} \in U^n$.

$\mathbb{N} \subseteq U$.

Proposition: $\{ y \mid (\exists\, n \geq 2)\, ((n, y) \in Q_\Sigma)\} \in NP_\Sigma{}^Q \setminus P_\Sigma{}^Q$.

gassnerc@uni-greifswald.de

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 $\boldsymbol{u} \in K_i$ within a time period bounded by $p_u(\lvert\boldsymbol{u}\rvert)$	$Q = Q_\Sigma = \cup_{i \geq 1} W_i$
K_1	≤ 1	$W_1 = \emptyset$
$\vdots$		
K_i	$\leq i$	$W_{i+1} = W_i \cup \{(i+1, \boldsymbol{u}) \mid \boldsymbol{u} \in K_i \ \& \ N_u^{W_i} \text{ rejects } \boldsymbol{u}\}$
$\vdots$		

$\Rightarrow N_u^{W_i}$ rejects $\boldsymbol{u}$ $\Leftrightarrow$ N_u^{Q} rejects $\boldsymbol{u}$.

gassnerc@uni-greifswald.de

An oracle Q with $P_\Sigma^Q \neq NP_\Sigma^Q$

Diagonalization techniques by Emerson

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

let $N_{\boldsymbol{u}}^B$ be the $P_\Sigma{}^B$-machine

- executing $p_{\boldsymbol{u}}(n)$ instructions of program $P_{\boldsymbol{u}}$ for any $\boldsymbol{x} \in U^n$.

$\mathbb{N} \subseteq U$.

$V_0 = \emptyset$.

Stage $i \geq 1$:

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

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

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

Proposition: $\{\, \boldsymbol{y} \mid (\exists\, n \geq 2)\, ((n,\ \boldsymbol{y}) \in Q_\Sigma)\} \in NP_\Sigma{}^Q \setminus P_\Sigma{}^Q$.

gassnerc@uni-greifswald.de

An oracle Q with $P_\Sigma^Q \neq NP_\Sigma^Q$

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_u^B be the P_Σ^B-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: $\{ y \mid (\exists\, n \geq 2)\, ((\alpha_n, y) \in Q_\Sigma)\} \in NP_\Sigma^Q \setminus P_\Sigma^Q$.

gassnerc@uni-greifswald.de

An oracle Q with $P_{\Sigma}^{Q} \neq NP_{\Sigma}^{Q}$

Diagonalization techniques (a generalization)

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

K_i	elements in a query on $\boldsymbol{u} \in K_i$ within a time period bounded by $p_{\boldsymbol{u}}(\lvert\boldsymbol{u}\rvert)$	$Q = Q_{\Sigma} = \cup_{i \geq 1} W_i$
K_1	$\notin \{\alpha_1, \alpha_2, \alpha_3, \ldots\}$	$W_1 = \emptyset$
$\vdots$		
K_i	$\notin \{\alpha_{i+1}, \alpha_{i+2}, \alpha_{i+3}, \ldots\}$	$W_{i+1} = W_i \cup \{(\alpha_{i+1}, \boldsymbol{u}) \mid \boldsymbol{u} \in K_i \;\&\; N_{\boldsymbol{u}}^{W_i} \text{ rejects } \boldsymbol{u}\}$
$\vdots$		

$\Rightarrow N_{\boldsymbol{u}}^{W_i}$ rejects $\boldsymbol{u}$ $\Leftrightarrow$ $N_{\boldsymbol{u}}^{Q}$ rejects $\boldsymbol{u}$.

gassnerc@uni-greifswald.de

An oracle Q with $P_\Sigma^Q \neq NP_\Sigma^Q$

Diagonalization techniques (a generalization)

3. If U is infinite, for suitable codes $\boldsymbol{u} \in U \subseteq U^\infty$ and any oracle $B \subseteq U^\infty$,

let $N_{\boldsymbol{u}}^B$ be the $P_\Sigma{}^B$-machine

- executing $p_{\boldsymbol{u}}(n)$ instructions of program $P_{\boldsymbol{u}}$ for any $\boldsymbol{x} \in U^n$.

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

$V_0 = \emptyset$.

Stage $i \geq 1$:

$K_i = \{\boldsymbol{u} \in U \mid (\forall j > i)(\forall B \subseteq U^\infty)$ $(N_{\boldsymbol{u}}^B$ does not compute or use the value α_j on $\boldsymbol{u})\}$

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

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

Proposition: $\{ \boldsymbol{y} \mid (\exists\, n \geq 2)\, ((\alpha_n, \boldsymbol{y}) \in Q_\Sigma)\} \in NP_\Sigma{}^Q \setminus P_\Sigma{}^Q$.

gassnerc@uni-greifswald.de

An oracle Q with $P_\Sigma^Q \neq NP_\Sigma^Q$

Using the undecidability of the Halting problem H_Σ

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

$\{\alpha_1, \alpha_2, \alpha_3, \ldots\} \subseteq U$ enumerable and decidable.

$$Q = Q_\Sigma = \{\ (\alpha_t, \boldsymbol{x}, Code(M))\ \mid$$

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

$$\&\ \ \underbrace{M(\boldsymbol{x})\downarrow^t}\}$$

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

Proposition: $H_\Sigma \in NP_\Sigma^Q \setminus P_\Sigma^Q$. $\qquad (P_\Sigma^Q \subseteq DEC_\Sigma)$

gassnerc@uni-greifswald.de

An oracle O_Σ with $P_\Sigma^{O_\Sigma} = NP_\Sigma^{O_\Sigma}$

A universal oracle:

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

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

gassnerc@uni-greifswald.de

An oracle O_Σ containing only tuples of length 1 with $\mathrm{P}_\Sigma^{O_\Sigma} = \mathrm{NP}_\Sigma^{O_\Sigma}$?

Structures over strings

$\Sigma = (U^*;\ \varepsilon, a, b, c_3, \ldots, c_u;\ \mathrm{add}, \mathrm{sub}_\mathrm{l}, \mathrm{sub}_\mathrm{r},\ f_1, \ldots, f_v;\ R_1, \ldots, R_w,\ =)$

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

$s = d_1 \cdots d_k \in U^*$ stored in one register

$d \in U$

$\mathrm{add}(s, d) = sd$ $\mathrm{sub}_\mathrm{l}(sd) = s$ $\mathrm{sub}_\mathrm{r}(sd) = d$

An oracle O_Σ 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

$$O_\Sigma = \{ (\underbrace{b,\ldots,b}_{t\,\times}, \boldsymbol{x}, Code(M)) \mid \boldsymbol{x} \in (U^*)^\infty \text{ \& } M \text{ is a non-deterministic } \Sigma\text{-machine using } O_\Sigma \text{ \& } M(\boldsymbol{x})\downarrow^t \}$$

$$Q_\Sigma = \{ (\underbrace{b\cdots b}_{t\,\times}, \boldsymbol{x}, Code(M)) \mid \boldsymbol{x} \in (U^*)^\infty \text{ \& } M \text{ is a deterministic } \Sigma\text{-machine \& } M(\boldsymbol{x})\downarrow^t \}$$

Theorem: There is **not** an oracle O with

$$b\cdots b\cdot \mathrm{string}(\boldsymbol{x})\cdot \mathrm{string}(Code(M)) \in O$$
$$\Leftrightarrow \quad \boldsymbol{x} \in (U^*)^\infty$$
$$\text{\& } M \text{ is a non-deterministic } \Sigma\text{-machine using } O$$
$$\text{\& } M(\boldsymbol{x})\downarrow^t .$$

No set !

Structures with P = NP

An additional relation R
on padded codes
of the members of a universal oracle O
with $P_{\Sigma}^{O} = NP_{\Sigma}^{O}$

P = NP for

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

P = NP for

Strings
with operations for adding and deleting the last character
(Gaßner, CiE 2007)

gassnerc@uni-greifswald.de

$\mathbb{Z}$ as oracle with $\mathrm{P}_{\mathbb{R}}^{\mathbb{Z}} \neq \mathrm{NP}_{\mathbb{R}}^{\mathbb{Z}}$

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

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

$\mathbb{Q} \in \mathrm{NP}_{\mathbb{R}}^{\mathbb{Z}}$.

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: $\mathrm{P}_{\mathbb{R}}^{\mathbb{Z}} \neq \mathrm{NP}_{\mathbb{R}}^{\mathbb{Z}}$.

gassnerc@uni-greifswald.de

$\mathbb{Z}$ as oracle with $P_{\mathbb{R}}^{\mathbb{Z}} \neq NP_{\mathbb{R}}^{\mathbb{Z}}$

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

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

$\mathbb{Q} \in NP_{\mathbb{R}}^{\mathbb{Z}}$.

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.

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

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

$$p_k(x) \in \mathbb{Z} \qquad p_k(x) \notin \mathbb{Z} \qquad p_k(x) \leq 0 \qquad p_k(x) < 0 \qquad (k \leq m).$$

$\Rightarrow$ 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 $q_i \to r$.

$\Rightarrow$ 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. $\Rightarrow$ ↯

Proposition: $P_{\mathbb{R}}^{\mathbb{Z}} \neq NP_{\mathbb{R}}^{\mathbb{Z}}$.

On Relativizations of the $P \stackrel{?}{=} NP$ Question for Several Structures

Thank you for your attention!

Christine Gaßner
Greifswald.

Thanks also to

Robert Bialowons,
Volkmar Liebscher,
Rainer Schimming.

gassnerc@uni-greifswald.de