Abstract computation over first-order structures

Universal BSS RAMs and their complexity

Christine Gaßner

University of Greifswald

Leeds 2025

Updated on October 25, 2025
(Two lines were changed.)

Outline

From algorithms over first-order structures to complexity

Motivation: Reasons for our generalization

Introduction: The BSS-RAM model

- Structures $\mathcal{A}$ of signature σ
- σ-Programs
- Transition systems and BSS RAMs
- An example

Complexity of BSS RAMs

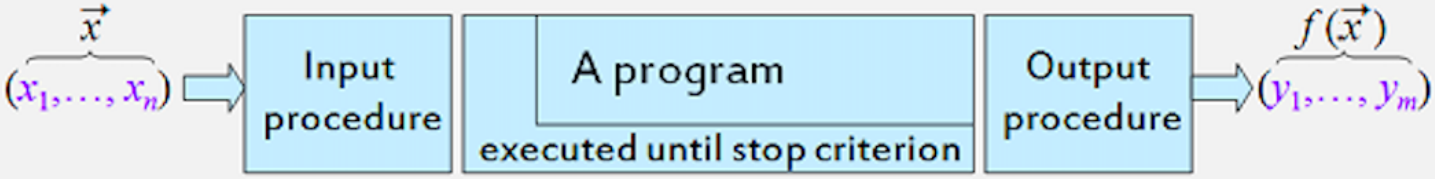
Interesting problems:
$\mathrm{P}_{\mathcal{A}}$-$\mathrm{NP}_{\mathcal{A}}$ problems and historical remarks

Universal machines and their importance

Our BSS-RAM model

With input and output procedure

Minsky (1961), Scott (1967), Blum/Shub/Smale (1989),... (offline):

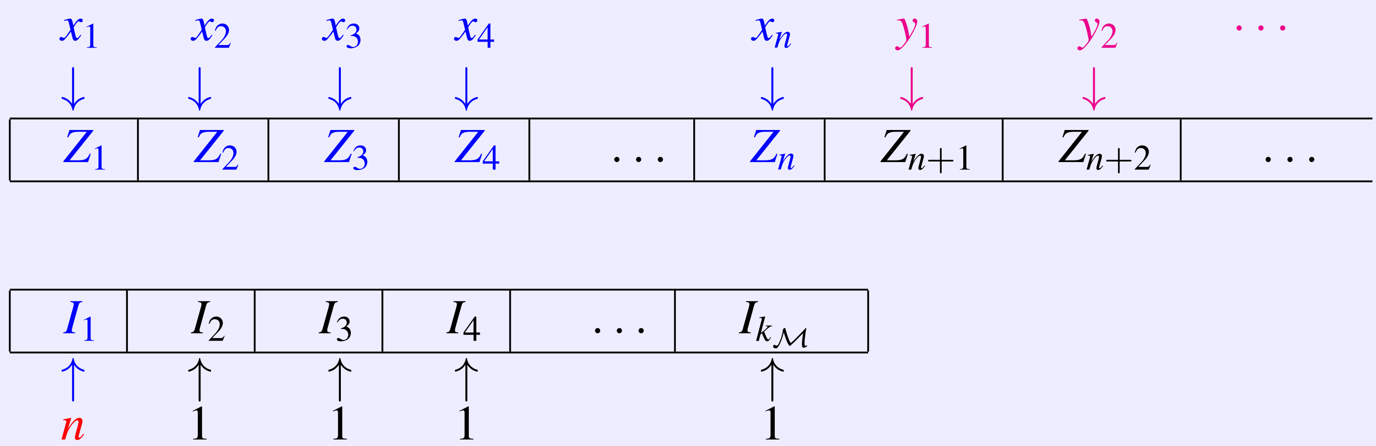


Uniform Computation over First-Order Structures

Input and output procedures for (digitally) non-deterministic BSS RAMs over $\mathcal{A}$

Input and output space: $U_{\mathcal{A}}^{\infty} =_{\text{df}} \bigcup_{i \geq 1} U_{\mathcal{A}}^{i}$

Input of $\vec{x} = (x_1, \ldots, x_n) \in U_{\mathcal{A}}^{\infty}$:



Note: Any machine can recognize the end of an input.

Output after a stop-instruction:

$(c(Z_1), \ldots, c(Z_{c(I_1)}))$

Algorithms over First-Order Structures

σ-programs for $[\text{in}]$finite $\mathcal{A}$-machines $\mathcal{M}$

- a structure $\mathcal{A} = (U_{\mathcal{A}}; c_1, \ldots, c_{n_1}; f_1, \ldots, f_{n_2}; r_1, \ldots, r_{n_3})$
- a finite signature $\sigma = (n_1; m_1, \ldots, m_{n_2}; k_1, \ldots, k_{n_3})$
- the first-order symbols $c_1^0, \ldots, c_{n_1}^0, f_1^{m_1}, \ldots, f_{n_2}^{m_{n_2}}, r_1^{k_1}, \ldots, r_{n_3}^{k_{n_3}}$

Definition (σ-program $\mathcal{P}_{\mathcal{M}}$ of an $\mathcal{A}$-machine)

1 : instruction$_1$; $\ldots$; $\ell_{\mathcal{P}_{\mathcal{M}}} - 1$: instruction$_{\ell_{\mathcal{P}_{\mathcal{M}}}-1}$; $\ell_{\mathcal{P}_{\mathcal{M}}}$: stop.

- Any instruction$_i$ is a σ-instruction.
- The execution of $\mathcal{P}_{\mathcal{M}}$ is:
  the stepwise transformation of configurations
  by the transition system $(\mathsf{S}_{\mathcal{M}}, \rightarrow_{\mathcal{M}})$.

Algorithms over First-Order Structures

σ-instructions for infinite $\mathcal{A}$-machines $\mathcal{M}$ and transport instructions

Z_1	Z_2	Z_3	Z_4	Z_5	$\dots$

Z-registers (for *individuals* in $U_{\mathcal{A}}$)

I_1	I_2	I_3	I_4	$\dots$	$I_{k_{\mathcal{M}}}$

Index registers (for *indices* in $\mathbb{N}_+$)

Computation	$\ell\colon\ Z_j := f_i^{m_i}(Z_{j_1}, \dots, Z_{j_{m_i}})$	(1)
	$\ell\colon\ Z_j := c_i^0$	(2)
Copy	$\ell\colon\ Z_{I_j} := Z_{I_k}$	(3)
Branching	$\ell\colon$ if $r_i^{k_i}(Z_{j_1}, \dots, Z_{j_{k_i}})$ then goto ℓ_1 else goto ℓ_2	(4)
Index	$\ell\colon\ I_j := 1$	(5)
	$\ell\colon\ I_j := I_j + 1$	(6)
	$\ell\colon$ if $I_j = I_k$ then goto ℓ_1 else goto ℓ_2	(7)
Stop	$l\colon$ stop	(8)

Overall States

Configurations of an $\mathcal{A}$-machine $\mathcal{M}$

Definition (Configurations in $S_{\mathcal{M}}$)

$$(\ell \,.\, \vec{\nu} \,.\, \bar{u}) \quad = \underbrace{(\ell, \nu_1, \ldots, \nu_{k_{\mathcal{M}}}, u_1, u_2, \ldots)}_{\text{a configuration of } \mathcal{M}}$$

B

I_1	I_2	I_3	$\ldots$	$I_{k_{\mathcal{M}}}$

index registers

pointers

read-and-write heads

Z_1	Z_2	Z_3	Z_4	Z_5	$\ldots$

Z-registers

tape

$\ell \in \mathcal{L}_{\mathcal{M}}$ $\quad \vec{\nu} = (\nu_1, \ldots, \nu_{k_{\mathcal{M}}}) \in \mathbb{N}_+^{k_{\mathcal{M}}}$ $\quad \bar{u} = (u_1, u_2, \ldots) \in U_{\mathcal{A}}^{\omega}$

- ℓ a label in $\mathcal{L}_{\mathcal{M}}$ — instruction counter, the internal state
- $\vec{\nu}$ indices of Z-registers — addresses, positions on the tape
- $\bar{u}$ a sequence of individuals

Unit Costs by a Computational System for $\mathcal{A}$-Machines

Transformation of configurations of an $\mathcal{A}$-machine $\mathcal{M}$

Definition (Transition system for any $\mathcal{A}$-machine $\mathcal{M}$)

$$\mathcal{S}_{\mathcal{M}} = (\mathsf{S}_{\mathcal{M}}, \to_{\mathcal{M}})$$

with a binary relation $\to_{\mathcal{M}} \subseteq \mathsf{S}^2_{\mathcal{M}}$

defined below.

(cf. Introduction 2020, ...; see also Börger for finite machines, ...)

Unit Costs by a Computational System for $\mathcal{A}$-Machines

Transformation of configurations of an $\mathcal{A}$-machine $\mathcal{M}$

$$\ell\colon\ Z_j := f_i^{m_i}(Z_{j_1}, \ldots, Z_{j_{m_i}})$$

$$(\ell \,.\, \vec{\nu} \,.\, \bar{u}) \quad \to_{\mathcal{M}} \quad (\ell + 1 \,.\, \vec{\nu} \,.\, (u_1, \ldots, u_{j-1}, f_i(u_{j_1}, \ldots, u_{j_{m_i}}), u_{j+1}, \ldots))$$

$$\ell\colon\ Z_j := c_i^0$$

$$(\ell \,.\, \vec{\nu} \,.\, \bar{u}) \quad \to_{\mathcal{M}} \quad (\ell + 1 \,.\, \vec{\nu} \,.\, (u_1, \ldots, u_{j-1}, c_i, u_{j+1}, \ldots))$$

$$\ell\colon\ Z_{I_j} := Z_{I_k}$$

$$(\ell \,.\, \vec{\nu} \,.\, \bar{u}) \quad \to_{\mathcal{M}} \quad (\ell + 1 \,.\, \vec{\nu} \,.\, (u_1, \ldots, u_{\nu_j - 1}, u_{\nu_k}, u_{\nu_j + 1}, \ldots))$$

$$\ell\colon\ \text{if } r_i^{k_i}(Z_{j_1}, \ldots, Z_{j_{k_i}}) \text{ then goto } \ell_1 \text{ else goto } \ell_2$$

$$(\ell \,.\, \vec{\nu} \,.\, \bar{u}) \quad \to_{\mathcal{M}} \quad (\ell_1 \,.\, \vec{\nu} \,.\, \bar{u}) \qquad \text{if } (u_{j_1}, \ldots, u_{j_{k_i}}) \in r_i$$

$$(\ell \,.\, \vec{\nu} \,.\, \bar{u}) \quad \to_{\mathcal{M}} \quad (\ell_2 \,.\, \vec{\nu} \,.\, \bar{u}) \qquad \text{if } (u_{j_1}, \ldots, u_{j_{k_i}}) \notin r_i$$

Unit Costs of $\mathcal{A}$-machines

Transformation of configurations of an $\mathcal{A}$-machine $\mathcal{M}$

ℓ: if $I_j = I_k$ then goto ℓ_1 else goto ℓ_2

$$(\ell \,.\, \vec{\nu} \,.\, \bar{u}) \rightarrow_{\mathcal{M}} (\ell_1 \,.\, \vec{\nu} \,.\, \bar{u}) \quad \text{if } \nu_j = \nu_k$$
$$(\ell \,.\, \vec{\nu} \,.\, \bar{u}) \rightarrow_{\mathcal{M}} (\ell_2 \,.\, \vec{\nu} \,.\, \bar{u}) \quad \text{if } \nu_j \neq \nu_k$$

ℓ: $I_j := 1$

$$(\ell \,.\, \vec{\nu} \,.\, \bar{u}) \rightarrow_{\mathcal{M}} (\ell + 1 \,.\, (\nu_1, \ldots, \nu_{j-1}, 1, \nu_{j+1}, \ldots) \,.\, \bar{u})$$

ℓ: $I_j := I_j + 1$

$$(\ell \,.\, \vec{\nu} \,.\, \bar{u}) \rightarrow_{\mathcal{M}} (\ell + 1 \,.\, (\nu_1, \ldots, \nu_{j-1}, \nu_j + 1, \nu_{j+1}, \ldots) \,.\, \bar{u})$$

$\ell_{\mathcal{P}}$: stop

$$(\ell_{\mathcal{P}} \,.\, \vec{\nu} \,.\, \bar{u}) \rightarrow_{\mathcal{M}} (\ell_{\mathcal{P}} \,.\, \vec{\nu} \,.\, \bar{u})$$

Semi-Decidable Decision Problems
Halting sets

Definition (Decision problems)

Any subset $P \subseteq U_{\mathcal{A}}^{\infty}$ (that contains all tuples) is a *decision problem*.

$\vec{x} \in U_{\mathcal{A}}^{\infty}$ $\mathcal{M}$ BSS RAM over $\mathcal{A}$

Definition (Halting process)

$\mathcal{M}$ *halts* on input $\vec{x}$ if $\ell_{\mathcal{P}}:$ stop is reached $\big[$for some guesses$\big]$.

$\mathcal{M}(\vec{x}) \downarrow^{t}$ if $\mathcal{M}$ halts on input $\vec{x}$ after t steps $\big[$for some guesses$\big]$.

$\mathcal{M}(\vec{x}) \downarrow$ if $\mathcal{M}(\vec{x}) \downarrow^{t}$ for some t.

$\mathcal{M}(\vec{x}) \uparrow$ if $\mathcal{M}(\vec{x}) \downarrow$ does not hold.

Definition (Halting set)

$H_{\mathcal{M}} = \{\vec{x} \in U_{\mathcal{A}}^{\infty} \mid \mathcal{M}(\vec{x}) \downarrow\}$ is the *halting set* of $\mathcal{M}$.

Properties (Uniform processing)

All inputs of any length can be (systematically) processed by one machine.

Definition (Unit costs)

One application of $\rightarrow_{\mathcal{M}}$ is *one step* of $\mathcal{M}$ and has cost 1.

Complexity of BSS RAMs
Time complexity

$\vec{x} = (x_1, \ldots, x_n) \in U_{\mathcal{A}}^{\infty}$
$\vec{y} = (y_1, \ldots, y_m) \in U_{\mathcal{A}}^{\infty}$

$$t_{\mathcal{M}}(\vec{x}) = \begin{cases} t & \text{if } \mathcal{M}(\vec{x}) \downarrow^{t} \\ \infty & \text{if } \mathcal{M}(\vec{x}) \uparrow \end{cases}$$ (for deterministic $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}$)

$$t_{\mathcal{M},\vec{x}}(\vec{y}) = \begin{cases} t & \text{if } \mathcal{M}(\vec{x}) \downarrow^{t} \\ \infty & \text{if } \mathcal{M}(\vec{x}) \uparrow \end{cases}$$ for the guesses $y_1, \ldots, y_m$

$$t_{\mathcal{M}}(\vec{x}) = \min_{\vec{y} \in U_{\mathcal{A}}^{\infty}} t_{\mathcal{M},\vec{x}}(\vec{y})$$ (for non-deterministic $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}^{\mathrm{ND}}$)

Definition (Time complexity $T_{\mathcal{M}} : \mathbb{N}_{+} \to \mathbb{N} \cup \{\infty\}$)

$$T_{\mathcal{M}}(n) = \max_{(x_1,\ldots,x_n) \in U_{\mathcal{A}}^{\infty}} \{ t_{\mathcal{M}}(x_1, \ldots, x_n) \mid \mathcal{M}(x_1, \ldots, x_n) \downarrow \}$$

Definition (Polynomial time semi-decidability)

$H_{\mathcal{M}}$ is *semi-decidable* over $\mathcal{A}$ in polynomial time if $\exists k(T_{\mathcal{M}} \in O(n^k))$.

Definition (The class $\mathrm{P}_{\mathcal{A}}$)

$\mathrm{P}_{\mathcal{A}} = \{P \subseteq U_{\mathcal{A}}^{\infty} \mid P$ is semi-decidable in polynomial time$\}$

$\mathrm{P}_{\mathcal{A}}$	class of all problems decidable in polynomial time
$\mathrm{DNP}_{\mathcal{A}}$	class of all problems digitally non-det. semi-decidable in polynomial time
$\mathrm{NP}_{\mathcal{A}}$	class of all problems nondeterministically semi-decidable in polynomial time

$\mathrm{P}_{\mathcal{A}}$-$\mathrm{NP}_{\mathcal{A}}$ Problems for First-Order Structures

Solved or unsolved

$\mathcal{A}$	$\mathrm{P}_{\mathcal{A}} \neq \mathrm{DNP}_{\mathcal{A}}$?	$\mathrm{DNP}_{\mathcal{A}} \neq \mathrm{NP}_{\mathcal{A}}$?
$\mathcal{A}_0 = (\{0,1\}; 1,0; ; =)$	?	no
$\mathbb{Z}_{\text{add}}^{=} = (\mathbb{Z}; \mathbb{Z}; +, -; =)$	yes (Th. 5 in [Gaßner '01])	yes, $2\mathbb{Z} \in \mathrm{NP}_{\mathcal{A}} \setminus \mathrm{DNP}_{\mathcal{A}}$
$\mathbb{Z}_{\text{add}}^{\leq} = (\mathbb{Z}; \mathbb{Z}; +, -; \leq)$	?	yes, $2\mathbb{Z} \in \mathrm{NP}_{\mathcal{A}} \setminus \mathrm{DNP}_{\mathcal{A}}$
$\mathbb{R}_{\text{add}}^{=} = (\mathbb{R}; \mathbb{R}; +, -; =)$	yes (Th. 4 in [Koiran '94])	no (Th. 2 in [Koiran '94])
$\mathbb{R}_{\text{add}}^{\leq} = (\mathbb{R}; \mathbb{R}; +, -; \leq)$	?	no (Th. 10 in [Koiran '94])
$\mathbb{R}_{\text{lin}}^{=} = (\mathbb{R}; 1; +, -, \{\phi_r \mid r \in \mathbb{R}\}; =)$	yes (vgl. [Meer '92])	no (Th. 14 in [Koiran '94])
$\mathbb{R}_{\text{lin}}^{\leq} = (\mathbb{R}; 1; +, -, \{\phi_r \mid r \in \mathbb{R}\}; \leq)$	?	no (Th. 17 in [Koiran '94])
$\mathbb{Z}^{\leq} = (\mathbb{Z}; \mathbb{Z}; +, -, \cdot; \leq)$	?	yes, $2\mathbb{Z} \in \mathrm{NP}_{\mathcal{A}} \setminus \mathrm{DNP}_{\mathcal{A}}$
$\mathbb{Q}^{\leq} = (\mathbb{Q}; \mathbb{Q}; +, -, \cdot; \leq)$	?	yes, $\mathbb{Q}^{\text{sq}} \in \mathrm{NP}_{\mathcal{A}} \setminus \mathrm{DNP}_{\mathcal{A}}$
$\mathbb{R}^{=} = (\mathbb{R}; \mathbb{R}; +, -, \cdot; =)$	?	yes, $\mathbb{R}_{\geq 0} \in \mathrm{NP}_{\mathcal{A}} \setminus \mathrm{DNP}_{\mathcal{A}}$
$\mathbb{R}^{\leq} = (\mathbb{R}; \mathbb{R}; +, -, \cdot; \leq)$	?	?
$\mathbb{C} = (\mathbb{C}; \mathbb{C}; +, -, \cdot; =)$	?	?

$\phi_r(x) = rx$,

$\mathbb{Q}^{\text{sq}} = \{q^2 \mid q \in \mathbb{Q}\}$, $\quad \mathbb{R}_{\geq 0} = \{r \in \mathbb{R} \mid r \geq 0\}$, $\quad 2\mathbb{Z} = \{2z \mid z \in \mathbb{Z}\}$.

See also Prunescu, ... The symbol ? means that I do not know more.

Known $\mathrm{P}_{\mathcal{A}}$-$\mathrm{NP}_{\mathcal{A}}$ Problems

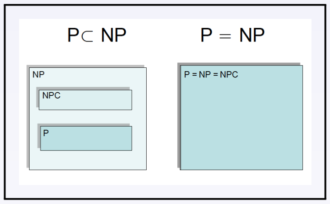
$\mathrm{NP}_{\mathcal{A}}$-complete problems

$$\mathcal{A}_0 =_{\mathrm{df}} (\{0,1\}; 0, 1; ; =)$$

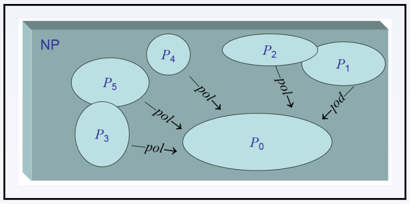
The problem P = NP? (a Millennium Prize Problem) is open.

SAT $\in$ P?

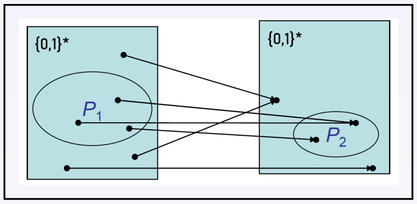
SAT is NP-complete (see Cook 1971) $\Rightarrow$ SAT $\in$ P implies P = NP.



Two cases



Reductions on P_0



One reduction of P_1 to P_2

$$\mathbb{R} =_{\mathrm{df}} (\mathbb{R}; \mathbb{R}; +, -, \cdot; <, =)$$

The problem $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$? is open.

$\mathrm{Uni}_0 \in \mathrm{P}_{\mathbb{R}}$?

Uni_0 is $\mathrm{NP}_{\mathbb{R}}$-hard (cf. BSS 1989) $\Rightarrow$ $\mathrm{Uni}_0 \in \mathrm{P}_{\mathbb{R}}$ implies $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$.

A Structure $\mathcal{A}$ with $\mathrm{P}_\mathcal{A} = \mathrm{NP}_\mathcal{A}$

A new question (Poizat 1995): Is there such a structure of finite signature?

Poizat's idea for answering his question: A new relation

2001 The recursive definition of a unary relation over trees derived from SAT without identity (G., Dagstuhl 2004)

2003 The complete proof (G., Preprint 2004/1 Greifswald)

2004 A structure of strings with identity fewer constructable (Hemmerling, Journal of Complexity 2005)

Embedding of structures to get $\mathrm{P}_\mathcal{A} = \mathrm{NP}_\mathcal{A}$

2005 derived from SAT for structures with two constants (G., Lyon 2005, CiE 2006)

2006 derived from SAT for structures with two computable individuals (G., CiE 2006)

2007 derived from Uni for structures with two constants (G., CiE 2007)

2008 derived from Uni for groups (with one constant) (G., CiE 2008)

Now derived from $\mathrm{Uni}_\mathcal{A}$ for structures $\mathcal{A}$ with or without constants (G., Preprint 2019/1 Greifswald, ...)

$\Rightarrow$ It is useful to consider the $\mathrm{NP}_\mathcal{A}$-complete problems $\mathrm{Uni}_\mathcal{A}$!

Complete Problems
over first-order structures

Properties (Certain halting problems)

$$Halt = \{\langle input, machine\rangle \subseteq U_{\mathcal{A}}^{\infty} \mid \langle machine\rangle \text{ halts on } \langle input\rangle\}$$

$$Uni = \{\langle input, machine, number\rangle \subseteq U_{\mathcal{A}}^{\infty} \mid \langle machine\rangle \text{ halts for } \langle input\rangle \text{ after } \langle number\rangle \text{ steps } \}$$

Properties (Sufficient conditions for $\mathrm{P}_{\mathcal{A}} = \mathrm{NP}_{\mathcal{A}}$)

Let

1. Uni is in $\mathrm{NP}_{\mathcal{A}}$ and
2. Uni is $\mathrm{NP}_{\mathcal{A}}$-complete.

If Uni is in $\mathrm{P}_{\mathcal{A}}$, then $\mathrm{P}_{\mathcal{A}} = \mathrm{NP}_{\mathcal{A}}$.

$\Rightarrow$ It is useful to consider universal BSS RAMs and their complexity.

Let $\mathcal{A} = (U_{\mathcal{A}}; (c_i)_{i\in N_1}; f_1, \ldots, f_{n_2}; r_1, \ldots, r_{n_3})$.

Here: Type 2 means encoding without constants.

Definition (Halting problems of type 2)

for machines without guessing, with binary guessing digits, with guessing

$$\mathbb{H}_{\mathcal{A}} = \{(\vec{a}^{\,(\mathcal{M},x_1)} \,.\, \vec{x} \,.\, \mathrm{code}_{n,x_n}(\mathcal{P}_{\mathcal{M}})) \in U_{\mathcal{A}}^{\infty} \mid \mathcal{M} \in \mathsf{M}_{\mathcal{A}} \quad \& \ \mathcal{M}(\vec{x}) \downarrow\}$$

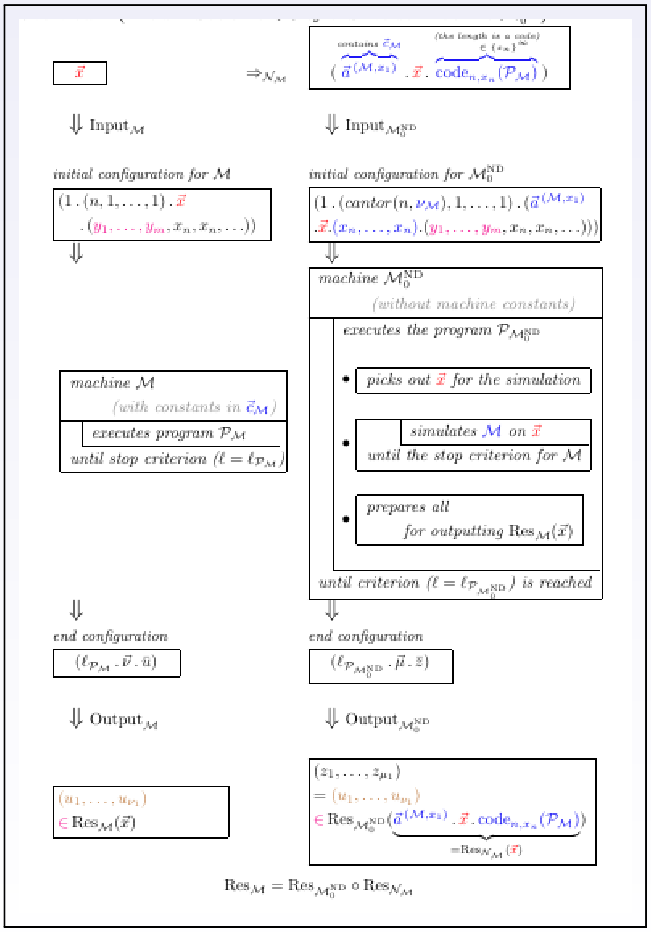
$$\mathbb{H}_{\mathcal{A}}^{\mathrm{DND}} = \{(\vec{a}^{\,(\mathcal{M},x_1)} \,.\, \vec{x} \,.\, \mathrm{code}_{n,x_n}(\mathcal{P}_{\mathcal{M}})) \in U_{\mathcal{A}}^{\infty} \mid \mathcal{M} \in \mathsf{M}_{\mathcal{A}}^{\mathrm{DND}} \ \& \ \mathcal{M}(\vec{x}) \downarrow\}$$

$$\mathbb{H}_{\mathcal{A}}^{\mathrm{ND}} = \{(\vec{a}^{\,(\mathcal{M},x_1)} \,.\, \vec{x} \,.\, \mathrm{code}_{n,x_n}(\mathcal{P}_{\mathcal{M}})) \in U_{\mathcal{A}}^{\infty} \mid \mathcal{M} \in \mathsf{M}_{\mathcal{A}}^{\mathrm{ND}} \quad \& \ \mathcal{M}(\vec{x}) \downarrow\}$$

Universal BSS RAMs

The simulation by a three-tape machine (cf. Introduction . . .)

Simulation of $\mathcal{M}$ by $\mathcal{M}_0^{[[\mathrm{D}]\mathrm{ND}]}$



Program of $\mathcal{M}_0^{[[\mathrm{D}]\mathrm{ND}]}$

The program $\mathcal{P}_0^{(3)}$

1 : $N := cantor_1(N_0)$; $C := cantor_2(N_0)$;
$L_P := \max\{m \mid p_m|C\}$; $(Z_{3,1}, \ldots, Z_{3,L_P}) := (Z_{1,1}, \ldots, Z_{1,L_P})$;
$(Z_{1,N_0+1}, \ldots, Z_{1,N_0+N}) := (Z_{1,L_P+1}, \ldots, Z_{1,L_P+N})$; $V := p_1^N$; $L := 1$;

ℓ_1 : if $L = L_P$ then goto ℓ_7 else goto ℓ_2;

ℓ_2 : $S := \max\{s \mid p_L^s|C\}$;
if $p_1|S$ then $E_1 := \max\{e \mid p_1^e|S\}$; . . . ;
if $p_{m_0+5}|S$ then $E_{m_0+5} := \max\{e \mid p_{m_0+5}^e|S\}$;
if $E_1 = 1$ or $E_1 = 2$ then $J_0 := N_0 + E_5$;
if $E_1 = 1$ or $E_1 = 4$ then $\{J_1 := N_0 + E_6; \ldots; J_{m_0} := N_0 + E_{m_0+5}\}$;
if $E_1 = 3$ or $E_1 = 5$ then $\{K_1 := \mathrm{comp}_{E_5}; K_2 := \mathrm{comp}_{E_6}\}$;
if $E_1 = 6$ or $E_1 = 7$ then $K_1 := \mathrm{comp}_{E_5}$;

ℓ_3 : if $E_1 = 1$ then goto $\tilde{\ell}_1$; . . . ; if $E_1 = 7$ then goto $\tilde{\ell}_7$;

$\tilde{\ell}_1$: if $E_2 = 1$ then $Z_{1,J_0} := f_1^{m_1}(Z_{1,J_1}, \ldots, Z_{1,J_{m_1}})$; . . . ;
if $E_2 = n_2$ then $Z_{1,J_0} := f_{n_2}^{m_{n_2}}(Z_{1,J_1}, \ldots, Z_{1,J_{m_{n_2}}})$; goto ℓ_4;

$\tilde{\ell}_2$: $Z_{1,J_0} := Z_{3,L}$; goto ℓ_4;

$\tilde{\ell}_3$: $Z_{1,K_1} := Z_{1,K_2}$; goto ℓ_4;

$\tilde{\ell}_4$: if $E_2 = 1$ & $r_1^{k_1}(Z_{1,J_1}, \ldots, Z_{1,J_{k_1}})$ then goto ℓ_5 else goto ℓ_6; . . . ;
if $E_2 = n_3$ & $r_{n_3}^{k_{n_3}}(Z_{1,J_1}, \ldots, Z_{1,J_{k_{n_3}}})$ then goto ℓ_5 else goto ℓ_6;

$\tilde{\ell}_5$: if $K_1 = K_2$ then goto ℓ_5 else goto ℓ_6;

$\tilde{\ell}_6$: if $K_1 > N_0 + 1$ then $V := V \div p_{E_5}^{K_1 - N_0 - 1}$; goto ℓ_4;

$\tilde{\ell}_7$: $V := V \cdot p_{E_5}$;

ℓ_4 : $L := L + 1$; goto ℓ_1;

ℓ_5 : $L := E_3$; goto ℓ_1;

ℓ_6 : $L := E_4$; goto ℓ_1;

ℓ_7 : $N := \max\{j \mid p_1^j|V\}$; $K_1 := 1$;

ℓ_8 : if $N \geq K_1$ then $\{Z_{1,K_1} := Z_{1,N_0+K_1}; K_1 := K_1 + 1;$ goto $\ell_8\}$; $N_0 := N$;

ℓ_9 : stop.

The subprogram $K_\nu := \mathrm{comp}_{E_\mu}$ (given in form of a pseudo instruction)

if $p_{E_\mu}|V$ then $K_\nu := N_0 + \max\{s \mid p_{E_\mu}^s|V\}$ else $\{V := V \cdot p_{E_\mu}; K_\nu := N_0 + 1\}$

Encoding BSS RAMs $\mathcal{M}$ for their Simulation

Gödel numbers for instructions

$p_1, p_2, \ldots$ the sequence of all prime numbers ($p_i \le p_{i+j}$ for $i, j \ge 1$)

$\mu_{\mathcal{P},\ell}$ code for the instruction with label ℓ in program $\mathcal{P}$

Type	e_1	e_2	e_3	e_4	e_5	e_6	e_7	$\cdots$	$e_{\ldots}$	$\mu_{\mathcal{P},\ell}$
(1)	1	i			j	j_1	j_2		j_{m_i}	$2^1 \cdot 3^i \cdot 11^j \cdot 13^{j_1} \cdot 17^{j_2} \cdots p_{m_i+5}^{j_{m_i}}$
(2)	2				j					$2^2 \cdot 11^j$
(3)	3				j	k				$2^3 \cdot 11^j \cdot 13^k$
(4)	4	i	ℓ_1	ℓ_2		j_1	j_2		j_{k_i}	$2^4 \cdot 3^i \cdot 5^{\ell_1} \cdot 7^{\ell_2} \cdot 13^{j_1} \cdot 17^{j_2} \cdots p_{k_i+5}^{j_{k_i}}$
(5)	5		ℓ_1	ℓ_2	j	k				$2^5 \cdot 5^{\ell_1} \cdot 7^{\ell_2} \cdot 11^j \cdot 13^k$
(6)	6				j					$2^6 \cdot 11^j$
(7)	7				j					$2^7 \cdot 11^j$
(8)	0									1

New Names for the Index Registers of $\mathcal{M}_0^{\mathrm{ND}}$

The content of these index registers

N_0	the length n_0 of the input/output of $\mathcal{M}_0^{\mathrm{ND}}$
N	the length n of the input of $\mathcal{M}$ (first, after the input)
C	the Gödel number $\nu_{\mathcal{M}}$
L	the label of the current instruction
L_P	$\ell_{\mathcal{M}}$
V	$p_{s_1}^{c(I_{s_1})} \cdots p_{s_\mu}^{c(I_{s_\mu})}$ for the registers $I_{s_1}, \ldots, I_{s_\mu}$ $(s_1 < \cdots < s_\mu \leq k_{\mathcal{M}})$ of $\mathcal{M}$
S	the code of the current instruction
$E_1, E_2, \ldots$	the current values $e_1, e_2, \ldots$
$J_0, J_1, \ldots$	the values $n_0 + e_5, n_0 + e_6, \ldots$
K_1	for computing indices $(n_0 + i)$
K_2	for computing indices $(n_0 + i)$

A Part of the Program of $\mathcal{M}_0^{\text{ND}}$

Evaluation of index registers and a case distinction

$1:$ $N := cantor_1(N_0);\ C := cantor_2(N_0);$
$L_P := \max\{m \mid p_m | C\};\ (Z_{3,1}, \ldots, Z_{3,L_P}) := (Z_{1,1}, \ldots, Z_{1,L_P});$
$(Z_{1,N_0+1}, \ldots, Z_{1,N_0+N}) := (Z_{1,L_P+1}, \ldots, Z_{1,L_P+N});\ V := p_1^N;\ L := 1;$

$\ell_1:$ if $L = L_P$ then goto ℓ_7 else goto ℓ_2;

$\ell_2:$ $S := \max\{s \mid p_L^s | C\};$
if $p_1 | S$ then $E_1 := \max\{e \mid p_1^e | S\};\ \ldots;$
if $p_{m_0+5} | S$ then $E_{m_0+5} := \max\{e \mid p_{m_0+5}^e | S\};$
if $E_1 = 1$ or $E_1 = 2$ then $J_0 := N_0 + E_5;$
if $E_1 = 1$ or $E_1 = 4$ then $\{J_1 := N_0 + E_6;\ \ldots;\ J_{m_0} := N_0 + E_{m_0+5}\};$
if $E_1 = 3$ or $E_1 = 5$ then $\{K_1 := \text{comp}_{E_5};\ K_2 := \text{comp}_{E_6}\};$
if $E_1 = 6$ or $E_1 = 7$ then $K_1 := \text{comp}_{E_5};$

$\ell_3:$ if $E_1 = 1$ then goto $\tilde{\ell}_1;\ \ldots;$ if $E_1 = 7$ then goto $\tilde{\ell}_7$;

$\tilde{\ell}_1:$ if $E_2 = 1$ then $Z_{1,J_0} := f_1^{m_1}(Z_{1,J_1}, \ldots, Z_{1,J_{m_1}});\ \ldots;$
if $E_2 = n_2$ then $Z_{1,J_0} := f_{n_2}^{m_{n_2}}(Z_{1,J_1}, \ldots, Z_{1,J_{m_{n_2}}});$ goto ℓ_4;

$\tilde{\ell}_2:$ $Z_{1,J_0} := Z_{3,L};$ goto ℓ_4;

$\tilde{\ell}_3:$ $Z_{1,K_1} := Z_{1,K_2};$ goto ℓ_4; $\ldots$

Examples (Pseudo index instructions and their complexity)

ℓ: $I_j := I_k \cdot I_m$
ℓ: if $I_k \leq I_m$ then goto ℓ_1 else $I_j := I_k - I_m$
ℓ: if $I_k < I_m$ then goto ℓ_1 else $I_j := I_k \div I_m$
ℓ: $I_j := cantor(I_k, I_{I_m})$
ℓ: $I_j := cantor_1(I_k)$
ℓ: $I_j := cantor_2(I_k)$
ℓ: if $I_m | I_k$ then goto ℓ_1 else else goto ℓ_2
ℓ: if $I_m | I_k$ then $I_j := \max\{j \mid I_m^j | I_k\}$ else goto ℓ_2
ℓ: $I_j := p_{I_k}$
ℓ: $I_j := p_i^{I_k}$

$\mu \div \nu = \max\{s \in \mathbb{N}_+ \mid s \cdot \nu \leq \mu\}$ for all $\nu, \mu \in \mathbb{N}_+$ with $\nu \leq \mu$

$\nu | \mu$ means that ν is a divisor of μ

$cantor_1(\mu) = \mu_1$ and $cantor_2(\mu) = \mu_2$ if $\mu = cantor(\mu_1, \mu_2)$

Thanks

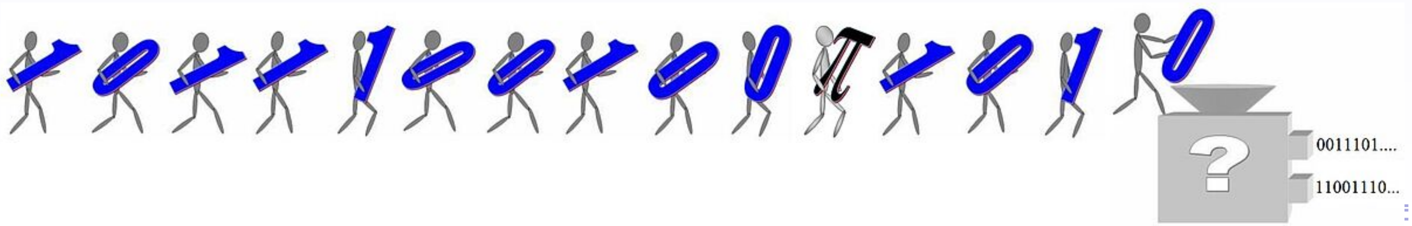
Thank you for your attention!

My thanks also go to the organizers.

I also thank
Patrick Steinbrunner and Sebastian Bierbaß,
Arno Pauly, Florian Steinberg,
Vasco Brattka, Philipp Schlicht, and Rupert Hölzl
for discussions.

My research was supported by many colleagues and
the International Office of the University Greifswald and the DVMLG.

I thank
Michael Schürmann, Volkmar Liebscher, Rainer Schimming, and
Michael Rathjen.



References

For more references and for P-NP problems see:

- Gaßner, C. *An introduction to a model of abstract computation: The BSS-RAM model*, Adrian Rezus (ed.): Contemporary Logic and Computing, College Publications [Landscapes in Logic 1], London 2020, pp. 574–603.
- Gaßner, C. *Abstract computation over first-order structures. Part I: Deterministic and non-deterministic BSS RAMs*, arXiv:2502.17539.
- Meer, K. *A note on a $P \neq NP$ result for a restricted class of real machines*, Journal of Complexity 8 (1992), 451–453.
- Koiran, P. *Computing over the reals with addition and order*, Theoretical Computer Science 133 (1994), 35–47.
- Gaßner, C. *The P-DNP problem for infinite abelian groups*, Journal of Complexity 17 (2001), 574–583.