Computation over Algebraic Structures and the Turing Reduction

Christine Gaßner

University Greifswald

CCC 2012 Trier

- Subject:
  BSS RAM model over any structure – a framework for study of
  - the abstract computability by machines over several structures
  - the uniform decidability over algebraic structures
  - the reducibility of problems

  on a high abstraction level
- Meaning:
  - allow to analyze the complexity of algorithms
  - better understanding the principles of object-oriented programming such as the encapsulation and the concept of virtual machines
  - improve the quality and the design of algorithms for computers
- Including:
  - several types of register machines
  - the Turing machine
  - the uniform BSS model of computation over the reals

- Subject:
  BSS RAM model over any structure – a framework for study of
  - the abstract computability by machines over several structures
  - the uniform decidability over algebraic structures
  - the reducibility of problems

  on a high abstraction level
- Meaning:
  - allow to analyze the complexity of algorithms
  - better understanding the principles of object-oriented programming such as the encapsulation and the concept of virtual machines
  - improve the quality and the design of algorithms for computers
- Including:
  - several types of register machines
  - the Turing machine
  - the uniform BSS model of computation over the reals

- Subject:
  BSS RAM model over any structure – a framework for study of
  - the abstract computability by machines over several structures
  - the uniform decidability over algebraic structures
  - the reducibility of problems

  on a high abstraction level
- Meaning:
  - allow to analyze the complexity of algorithms
  - better understanding the principles of object-oriented programming such as the encapsulation and the concept of virtual machines
  - improve the quality and the design of algorithms for computers
- Including:
  - several types of register machines
  - the Turing machine
  - the uniform BSS model of computation over the reals

- Subject:
  BSS RAM model over any structure – a framework for study of
  - the abstract computability by machines over several structures
  - the uniform decidability over algebraic structures
  - the reducibility of problems

  on a high abstraction level
- Meaning:
  - allow to analyze the complexity of algorithms
  - better understanding the principles of object-oriented programming such as the encapsulation and the concept of virtual machines
  - improve the quality and the design of algorithms for computers
- Including:
  - several types of register machines
  - the Turing machine
  - the uniform BSS model of computation over the reals

Outline

- The model
  - machines over algebraic structures
- Turing reductions
  - computed by machines over algebraic structures
- A hierarchy
  - derived from the arithmetical hierarchy
- A first characterization of the class Δ_2^0
  - the Limit Lemma
- The transfer of a further theorem from the Recursion Theory
  - a generalization of the Friedberg-Muchnik Theorem

Outline

- The model
  - machines over algebraic structures
- Turing reductions
  - computed by machines over algebraic structures
- A hierarchy
  - derived from the arithmetical hierarchy
- A first characterization of the class Δ_2^0
  - the Limit Lemma
- The transfer of a further theorem from the Recursion Theory
  - a generalization of the Friedberg-Muchnik Theorem

Computation over $\mathcal{A} = (U; (d_j)_{j \in J_0}; (f_j)_{j \in J_1}; (R_j)_{j \in J_2}, =)$.

- Computation instructions:

  $l\colon\ Z_j := f_k(Z_{j_1}, \ldots, Z_{j_{m_k}})$ (e.g. $l\colon\ Z_j := Z_{j_1} + Z_{j_2}$),

  $l\colon\ Z_j := d_k$,

- Branching instructions:

  $l\colon\ \text{if } Z_i = Z_j \text{ then goto } l_1 \text{ else goto } l_2$,

  $l\colon\ \text{if } R_k(Z_{j_1}, \ldots, Z_{j_{n_k}}) \text{ then goto } l_1 \text{ else goto } l_2$,

- Copy instructions:

  $l\colon\ Z_{I_j} := Z_{I_k}$,

- Index instructions:

  $l\colon\ I_j := 1$,

  $l\colon\ I_j := I_j + 1$,

  $l\colon\ \text{if } I_j = I_k \text{ then goto } l_1 \text{ else goto } l_2$.

Computation over $\mathcal{A} = (U; (d_j)_{j\in J_0}; (f_j)_{j\in J_1}; (R_j)_{j\in J_2}, =)$.

- Computation instructions:
  $l\colon\ Z_j := f_k(Z_{j_1}, \ldots, Z_{j_{m_k}})$ (e.g. $l\colon\ Z_j := Z_{j_1} + Z_{j_2}$),
  $l\colon\ Z_j := d_k$,
- Branching instructions:
  $l\colon$ if $Z_i = Z_j$ then goto l_1 else goto l_2,
  $l\colon$ if $R_k(Z_{j_1}, \ldots, Z_{j_{n_k}})$ then goto l_1 else goto l_2,
- Copy instructions:
  $l\colon\ Z_{I_j} := Z_{I_k}$,
- Index instructions:
  $l\colon\ I_j := 1$,
  $l\colon\ I_j := I_j + 1$,
  $l\colon$ if $I_j = I_k$ then goto l_1 else goto l_2.

Computation over $\mathcal{A} = (U; (d_j)_{j \in J_0}; (f_j)_{j \in J_1}; (R_j)_{j \in J_2}, =)$.

- Computation instructions:
  $l\colon\ Z_j := f_k(Z_{j_1}, \ldots, Z_{j_{m_k}})$ (e.g. $l\colon\ Z_j := Z_{j_1} + Z_{j_2}$),
  $l\colon\ Z_j := d_k$,
- Branching instructions:
  $l\colon$ if $Z_i = Z_j$ then goto l_1 else goto l_2,
  $l\colon$ if $R_k(Z_{j_1}, \ldots, Z_{j_{n_k}})$ then goto l_1 else goto l_2,
- Copy instructions:
  $l\colon\ Z_{I_j} := Z_{I_k}$,
- Index instructions:
  $l\colon\ I_j := 1$,
  $l\colon\ I_j := I_j + 1$,
  $l\colon$ if $I_j = I_k$ then goto l_1 else goto l_2.

Computation over $\mathcal{A} = (U; (d_j)_{j\in J_0}; (f_j)_{j\in J_1}; (R_j)_{j\in J_2}, =)$.

- Computation instructions:
  $l\colon\ Z_j := f_k(Z_{j_1}, \ldots, Z_{j_{m_k}})$ (e.g. $l\colon\ Z_j := Z_{j_1} + Z_{j_2}$),
  $l\colon\ Z_j := d_k$,
- Branching instructions:
  $l\colon\ \text{if } Z_i = Z_j \text{ then goto } l_1 \text{ else goto } l_2$,
  $l\colon\ \text{if } R_k(Z_{j_1}, \ldots, Z_{j_{n_k}}) \text{ then goto } l_1 \text{ else goto } l_2$,
- Copy instructions:
  $l\colon\ Z_{I_j} := Z_{I_k}$,
- Index instructions:
  $l\colon\ I_j := 1$,
  $l\colon\ I_j := I_j + 1$,
  $l\colon\ \text{if } I_j = I_k \text{ then goto } l_1 \text{ else goto } l_2$.

- Input and output space: $U^\infty =_{\mathrm{df}} \bigcup_{i \geq 1} U^i$
- Input of $\vec{x} = (x_1, \ldots, x_n) \in U^\infty$:

$$Z_1 := x_1; Z_2 := x_2; \ldots; Z_n := x_n;$$
$$Z_{n+1} := x_n; Z_{n+2} := x_n; \ldots$$
$$I_1 := n;$$

- Output of $Z_1, \ldots, Z_{I_1}$.

- $\mathsf{M}_{\mathcal{A}}$ – machines over $\mathcal{A}$
- $\mathsf{M}_{\mathcal{A}}(\mathcal{O})$ – machines using $\mathcal{O} \subseteq U^\infty$ as oracle

Oracle instructions:

l: if $(Z_1, \ldots, Z_{I_1}) \in \mathcal{O}$ then goto l_1 else goto l_2.

- Input and output space: $U^\infty =_{\rm df} \bigcup_{i\geq 1} U^i$
- Input of $\vec{x} = (x_1, \ldots, x_n) \in U^\infty$:

$$Z_1 := x_1; Z_2 := x_2; \ldots; Z_n := x_n;$$
$$Z_{n+1} := x_n; Z_{n+2} := x_n; \ldots$$
$$I_1 := n;$$

- Output of $Z_1, \ldots, Z_{I_1}$.

- $\mathrm{M}_{\mathcal{A}}$ – machines over $\mathcal{A}$
- $\mathrm{M}_{\mathcal{A}}(\mathcal{O})$ – machines using $\mathcal{O} \subseteq U^\infty$ as oracle

Oracle instructions:

l: if $(Z_1, \ldots, Z_{I_1}) \in \mathcal{O}$ then goto l_1 else goto l_2.

- Input and output space: $U^\infty =_{\mathrm{df}} \bigcup_{i \geq 1} U^i$
- Input of $\vec{x} = (x_1, \ldots, x_n) \in U^\infty$:

$$Z_1 := x_1; Z_2 := x_2; \ldots; Z_n := x_n;$$
$$Z_{n+1} := x_n; Z_{n+2} := x_n; \ldots$$
$$I_1 := n;$$

- Output of $Z_1, \ldots, Z_{I_1}$.

- $\mathrm{M}_{\mathcal{A}}$ – machines over $\mathcal{A}$
- $\mathrm{M}_{\mathcal{A}}(\mathcal{O})$ – machines using $\mathcal{O} \subseteq U^\infty$ as oracle

Oracle instructions:

l: if $(Z_1, \ldots, Z_{I_1}) \in \mathcal{O}$ then goto l_1 else goto l_2.

- Input and output space: $U^\infty =_{\mathrm{df}} \bigcup_{i\geq 1} U^i$
- Input of $\vec{x} = (x_1, \ldots, x_n) \in U^\infty$:

$$Z_1 := x_1; Z_2 := x_2; \ldots; Z_n := x_n;$$
$$Z_{n+1} := x_n; Z_{n+2} := x_n; \ldots$$
$$I_1 := n;$$

- Output of $Z_1, \ldots, Z_{I_1}$.
- $\mathrm{M}_{\mathcal{A}}$ – machines over $\mathcal{A}$
- $\mathrm{M}_{\mathcal{A}}(\mathcal{O})$ – machines using $\mathcal{O} \subseteq U^\infty$ as oracle

Oracle instructions:

l: if $(Z_1, \ldots, Z_{I_1}) \in \mathcal{O}$ then goto l_1 else goto l_2.

- Input and output space: $U^\infty =_{\mathrm{df}} \bigcup_{i\geq 1} U^i$
- Input of $\vec{x} = (x_1, \ldots, x_n) \in U^\infty$:

$$Z_1 := x_1; Z_2 := x_2; \ldots; Z_n := x_n;$$
$$Z_{n+1} := x_n; Z_{n+2} := x_n; \ldots$$
$$I_1 := n;$$

- Output of $Z_1, \ldots, Z_{I_1}$.
- $\mathsf{M}_\mathcal{A}$ – machines over $\mathcal{A}$
- $\mathsf{M}_\mathcal{A}(\mathcal{O})$ – machines using $\mathcal{O} \subseteq U^\infty$ as oracle

Oracle instructions:

l: if $(Z_1, \ldots, Z_{I_1}) \in \mathcal{O}$ then goto l_1 else goto l_2.

- $\mathbb{H}_{\mathcal{A}} = \{(\vec{x} \,.\, c_{\mathcal{M}}) \mid \vec{x} \in U^{\infty} \;\&\; \mathcal{M} \in \mathsf{M}_{\mathcal{A}} \;\&\; \mathcal{M}(\vec{x}) \downarrow\}$
- where

$$\begin{aligned} \vec{x} &= (x_1, \ldots, x_n) \\ c_{\mathcal{M}} &= \text{code}(\mathcal{M}) = (s_1, \ldots, s_m) \\ (\vec{x} \,.\, c_{\mathcal{M}}) &= (x_1, \ldots, x_n, s_1, \ldots, s_m) \end{aligned}$$

$$\mathcal{M}(\vec{x}) \downarrow \;\hat{=}\; \mathcal{M} \text{ halts on } \vec{x}$$

- $\mathbb{H}_{\mathcal{A}} \in \text{REC}_{\mathcal{A}}$ if $\mathcal{A}$ is a structure of finite signature
  $\mathbb{H}_{\mathcal{A}} \notin \text{DEC}_{\mathcal{A}}$

  $\text{REC}_{\mathcal{A}}$ – recognizable (semi-decidable) problems
  $\text{DEC}_{\mathcal{A}}$ – decidable problems

- $\mathbb{H}^{\mathcal{O}}_{\mathcal{A}} = \{(\vec{x} \,.\, c_{\mathcal{M}}) \mid \vec{x} \in U^{\infty} \;\&\; \mathcal{M} \in \mathsf{M}_{\mathcal{A}}(\mathcal{O}) \;\&\; \mathcal{M}(\vec{x}) \downarrow\}$
- $\mathbb{H}^{\mathcal{O}}_{\mathcal{A}} \notin \text{DEC}^{\mathcal{O}}_{\mathcal{A}}$

- $\mathbb{H}_{\mathcal{A}} = \{(\vec{x}\,.\,c_{\mathcal{M}}) \mid \vec{x} \in U^{\infty} \;\&\; \mathcal{M} \in \mathsf{M}_{\mathcal{A}} \;\&\; \mathcal{M}(\vec{x})\downarrow\}$
- where

$$\begin{aligned} \vec{x} &= (x_1, \ldots, x_n) \\ c_{\mathcal{M}} &= \text{code}(\mathcal{M}) = (s_1, \ldots, s_m) \\ (\vec{x}\,.\,c_{\mathcal{M}}) &= (x_1, \ldots, x_n, s_1, \ldots, s_m) \end{aligned}$$

$$\mathcal{M}(\vec{x})\downarrow \;\hat{=}\; \mathcal{M} \text{ halts on } \vec{x}$$

- $\mathbb{H}_{\mathcal{A}} \in \mathrm{REC}_{\mathcal{A}}$ if $\mathcal{A}$ is a structure of finite signature
  $\mathbb{H}_{\mathcal{A}} \notin \mathrm{DEC}_{\mathcal{A}}$

  $\mathrm{REC}_{\mathcal{A}}$ – recognizable (semi-decidable) problems
  $\mathrm{DEC}_{\mathcal{A}}$ – decidable problems

- $\mathbb{H}^{\mathcal{O}}_{\mathcal{A}} = \{(\vec{x}\,.\,c_{\mathcal{M}}) \mid \vec{x} \in U^{\infty} \;\&\; \mathcal{M} \in \mathsf{M}_{\mathcal{A}}(\mathcal{O}) \;\&\; \mathcal{M}(\vec{x})\downarrow\}$
- $\mathbb{H}^{\mathcal{O}}_{\mathcal{A}} \notin \mathrm{DEC}^{\mathcal{O}}_{\mathcal{A}}$

- $\mathbb{H}_{\mathcal{A}} = \{(\vec{x}\,.\,c_{\mathcal{M}}) \mid \vec{x} \in U^{\infty} \;\&\; \mathcal{M} \in \mathsf{M}_{\mathcal{A}} \;\&\; \mathcal{M}(\vec{x}) \downarrow\}$
- where

$$\begin{array}{rcl} \vec{x} & = & (x_1, \ldots, x_n) \\ c_{\mathcal{M}} & = & \text{code}(\mathcal{M}) = (s_1, \ldots, s_m) \\ (\vec{x}\,.\,c_{\mathcal{M}}) & = & (x_1, \ldots, x_n, s_1, \ldots, s_m) \\ & & \\ \mathcal{M}(\vec{x}) \downarrow & \hat{=} & \mathcal{M} \text{ halts on } \vec{x} \end{array}$$

- $\mathbb{H}_{\mathcal{A}} \in \text{REC}_{\mathcal{A}}$ if $\mathcal{A}$ is a structure of finite signature
  $\mathbb{H}_{\mathcal{A}} \notin \text{DEC}_{\mathcal{A}}$

  $\text{REC}_{\mathcal{A}}$ – recognizable (semi-decidable) problems
  $\text{DEC}_{\mathcal{A}}$ – decidable problems

- $\mathbb{H}^{\mathcal{O}}_{\mathcal{A}} = \{(\vec{x}\,.\,c_{\mathcal{M}}) \mid \vec{x} \in U^{\infty} \;\&\; \mathcal{M} \in \mathsf{M}_{\mathcal{A}}(\mathcal{O}) \;\&\; \mathcal{M}(\vec{x}) \downarrow\}$
- $\mathbb{H}^{\mathcal{O}}_{\mathcal{A}} \notin \text{DEC}^{\mathcal{O}}_{\mathcal{A}}$

- $\mathbb{H}_{\mathcal{A}} = \{(\vec{x}\,.\,c_{\mathcal{M}}) \mid \vec{x} \in U^{\infty} \;\&\; \mathcal{M} \in \mathsf{M}_{\mathcal{A}} \;\&\; \mathcal{M}(\vec{x}) \downarrow\}$
- where

$$\begin{aligned} \vec{x} &= (x_1, \ldots, x_n) \\ c_{\mathcal{M}} &= \text{code}(\mathcal{M}) = (s_1, \ldots, s_m) \\ (\vec{x}\,.\,c_{\mathcal{M}}) &= (x_1, \ldots, x_n, s_1, \ldots, s_m) \\ \\ \mathcal{M}(\vec{x}) \downarrow &\;\hat{=}\; \mathcal{M} \text{ halts on } \vec{x} \end{aligned}$$

- $\mathbb{H}_{\mathcal{A}} \in \text{REC}_{\mathcal{A}}$ if $\mathcal{A}$ is a structure of finite signature
  $\mathbb{H}_{\mathcal{A}} \notin \text{DEC}_{\mathcal{A}}$

  $\text{REC}_{\mathcal{A}}$ – recognizable (semi-decidable) problems
  $\text{DEC}_{\mathcal{A}}$ – decidable problems

- $\mathbb{H}^{\mathcal{O}}_{\mathcal{A}} = \{(\vec{x}\,.\,c_{\mathcal{M}}) \mid \vec{x} \in U^{\infty} \;\&\; \mathcal{M} \in \mathsf{M}_{\mathcal{A}}(\mathcal{O}) \;\&\; \mathcal{M}(\vec{x}) \downarrow\}$
- $\mathbb{H}^{\mathcal{O}}_{\mathcal{A}} \notin \text{DEC}^{\mathcal{O}}_{\mathcal{A}}$

The Turing Reduction
over Structures $\mathcal{A}$

- $P, Q \subseteq \bigcup_{i \geq 1} U^i$

  $P \preceq_T Q$ — P is *easier than* Q, P is decidable by a machine in $\mathsf{M}_{\mathcal{A}}(Q)$.

  $P \precneqq_T Q$ — P is *strictly easier than* Q, Q cannot be decided by a machine in $\mathsf{M}_{\mathcal{A}}(P)$.

- $\Rightarrow$ For the Halting Problem:

$$P \in \mathrm{REC}_{\mathcal{A}} \Rightarrow P \preceq_1 \mathbb{H}_{\mathcal{A}} \text{ (one-one reduction over } \mathcal{A}\text{)}$$
$$\Rightarrow P \preceq_T \mathbb{H}_{\mathcal{A}}$$

- $P, Q \subseteq \bigcup_{i \geq 1} U^i$

$P \preceq_T Q$ — P is *easier than* Q,
P is decidable by a machine in $\mathsf{M}_{\mathcal{A}}(Q)$.

$P \precneqq_T Q$ — P is *strictly easier than* Q,
Q cannot be decided by a machine in $\mathsf{M}_{\mathcal{A}}(P)$.

- $\Rightarrow$ For the Halting Problem:

$$P \in \mathrm{REC}_{\mathcal{A}} \Rightarrow P \preceq_1 \mathbb{H}_{\mathcal{A}} \text{ (one-one reduction over } \mathcal{A}\text{)}$$
$$\Rightarrow P \preceq_T \mathbb{H}_{\mathcal{A}}$$

- $P, Q \subseteq \bigcup_{i \geq 1} U^i$

  $P \preceq_T Q$ P is *easier than* Q,
  P is decidable by a machine in $\mathsf{M}_{\mathcal{A}}(Q)$.

  $P \precneqq_T Q$ P is *strictly easier than* Q,
  Q cannot be decided by a machine in $\mathsf{M}_{\mathcal{A}}(P)$.

- $\Rightarrow$ For the Halting Problem:

$$P \in \mathrm{REC}_{\mathcal{A}} \Rightarrow P \preceq_1 \mathbb{H}_{\mathcal{A}} \text{ (one-one reduction over } \mathcal{A}\text{)}$$
$$\Rightarrow P \preceq_T \mathbb{H}_{\mathcal{A}}$$

A Hierarchy
(Analogously to the Arithmetical Hierarchy)

- Definition (semantically by deterministic machines):
  $\mathcal{A}$ is fixed.

$$
\begin{aligned}
\Sigma_0^0 &= \mathrm{DEC}_{\mathcal{A}}, \\
\Pi_n^0 &= \{U^\infty \setminus P \mid P \in \Sigma_n^0\}, \\
\Delta_n^0 &= \Sigma_n^0 \cap \Pi_n^0, \\
\Sigma_{n+1}^0 &= \{P \subseteq U^\infty \mid (\exists Q \in \Sigma_n^0)(P \preceq_1 \mathbb{H}_{\mathcal{A}}^Q)\}.
\end{aligned}
$$

- The first level:

$$
\begin{aligned}
\Sigma_1^0 &= \mathrm{REC}_{\mathcal{A}} &&= \{P \subseteq U^\infty \mid P \preceq_1 \mathbb{H}_{\mathcal{A}}\}, \\
\Pi_1^0 & &&= \{P \subseteq U^\infty \mid P \preceq_1 U^\infty \setminus \mathbb{H}_{\mathcal{A}}\}, \\
\Delta_1^0 &= \mathrm{DEC}_{\mathcal{A}} &&= \{P \subseteq U^\infty \mid P \preceq_T \emptyset\},
\end{aligned}
$$

A Hierarchy
(Analogously to the Arithmetical Hierarchy)

- Definition (semantically by deterministic machines):
  $\mathcal{A}$ is fixed.

$$\begin{aligned}
\Sigma_0^0 &= \mathrm{DEC}_{\mathcal{A}}, \\
\Pi_n^0 &= \{U^\infty \setminus P \mid P \in \Sigma_n^0\}, \\
\Delta_n^0 &= \Sigma_n^0 \cap \Pi_n^0, \\
\Sigma_{n+1}^0 &= \{P \subseteq U^\infty \mid (\exists Q \in \Sigma_n^0)(P \preceq_1 \mathbb{H}_{\mathcal{A}}^Q)\}.
\end{aligned}$$

- The first level:

$$\begin{aligned}
\Sigma_1^0 &= \mathrm{REC}_{\mathcal{A}} &&= \{P \subseteq U^\infty \mid P \preceq_1 \mathbb{H}_{\mathcal{A}}\}, \\
\Pi_1^0 & &&= \{P \subseteq U^\infty \mid P \preceq_1 U^\infty \setminus \mathbb{H}_{\mathcal{A}}\}, \\
\Delta_1^0 &= \mathrm{DEC}_{\mathcal{A}} &&= \{P \subseteq U^\infty \mid P \preceq_T \emptyset\}.
\end{aligned}$$

A Hierarchy
(Analogously to the Arithmetical Hierarchy)

- Definition (semantically by deterministic machines):
  $\mathcal{A}$ is fixed.

$$
\begin{array}{lcl}
\Sigma_0^0 & = & \mathrm{DEC}_{\mathcal{A}}, \\
\Pi_n^0 & = & \{U^\infty \setminus P \mid P \in \Sigma_n^0\}, \\
\Delta_n^0 & = & \Sigma_n^0 \cap \Pi_n^0, \\
\Sigma_{n+1}^0 & = & \{P \subseteq U^\infty \mid (\exists Q \in \Sigma_n^0)(P \preceq_1 \mathbb{H}_{\mathcal{A}}^Q)\}.
\end{array}
$$

- The first level:

$$
\begin{array}{lcllcl}
\Sigma_1^0 & = & \mathrm{REC}_{\mathcal{A}} & = & \{P \subseteq U^\infty \mid P \preceq_1 \mathbb{H}_{\mathcal{A}}\}, \\
\Pi_1^0 & & & = & \{P \subseteq U^\infty \mid P \preceq_1 U^\infty \setminus \mathbb{H}_{\mathcal{A}}\}, \\
\Delta_1^0 & = & \mathrm{DEC}_{\mathcal{A}} & = & \{P \subseteq U^\infty \mid P \preceq_T \emptyset\},
\end{array}
$$

A Hierarchy
(Analogously to the Arithmetical Hierarchy)

- Definition (semantically by deterministic machines):
  $\mathcal{A}$ is fixed.

$$\begin{array}{rcl}
\Sigma_0^0 & = & \mathrm{DEC}_{\mathcal{A}}, \\
\Pi_n^0 & = & \{U^\infty \setminus P \mid P \in \Sigma_n^0\}, \\
\Delta_n^0 & = & \Sigma_n^0 \cap \Pi_n^0, \\
\Sigma_{n+1}^0 & = & \{P \subseteq U^\infty \mid (\exists Q \in \Sigma_n^0)(P \preceq_1 \mathbb{H}_{\mathcal{A}}^{Q})\}.
\end{array}$$

- The second level:

$$\begin{array}{rcccl}
\Sigma_2^0 & = & \mathrm{REC}_{\mathcal{A}}^{\mathbb{H}_{\mathcal{A}}} & = & \{P \subseteq U^\infty \mid P \preceq_1 \mathbb{H}_{\mathcal{A}}^{\mathbb{H}_{\mathcal{A}}}\}, \\
\Pi_2^0 & & & = & \{P \subseteq U^\infty \mid P \preceq_1 U^\infty \setminus \mathbb{H}_{\mathcal{A}}^{\mathbb{H}_{\mathcal{A}}}\}, \\
\Delta_2^0 & = & \mathrm{DEC}_{\mathcal{A}}^{\mathbb{H}_{\mathcal{A}}} & = & \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_{\mathcal{A}}\}.
\end{array}$$

A Characterization of $\Delta^0_2 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_{\mathcal{A}}\}$

- Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.
- χ_P – the characteristic function of the problem P.
- Let $P \subseteq U^\infty$.
  (1) $P \in \Delta^0_2$.
  (2) There is a computable function $g : U^\infty \to \{0, 1\}$
  defined on $\{(n \,.\, \vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$
  such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s \,.\, \vec{x})$.

Lemma (First Part of Limit Lemma)

If (1), then (2).

Lemma (Second Part of Limit Lemma)

If (2), then (1).

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

- Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.
- χ_P – the characteristic function of the problem P.
- Let $P \subseteq U^\infty$.
  (1) $P \in \Delta_2^0$.
  (2) There is a computable function $g : U^\infty \to \{0, 1\}$
  defined on $\{(n\,.\,\vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$
  such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s\,.\,\vec{x})$.

Lemma (First Part of Limit Lemma)

If (1), then (2).

Lemma (Second Part of Limit Lemma)

If (2), then (1).

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

- Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.
- χ_P – the characteristic function of the problem P.
- Let $P \subseteq U^\infty$.

  (1) $P \in \Delta_2^0$.

  (2) There is a computable function $g : U^\infty \to \{0, 1\}$
  defined on $\{(n \,.\, \vec{x}) \mid n \in \mathbb{N} \,\&\, \vec{x} \in U^\infty\}$
  such that $\chi_P(\vec{x}) = \lim_{s \to \infty} g(s \,.\, \vec{x})$.

Lemma (First Part of Limit Lemma)

If (1), then (2).

Lemma (Second Part of Limit Lemma)

If (2), then (1).

A Characterization of $\Delta^0_2 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

- Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.
- χ_P – the characteristic function of the problem P.
- Let $P \subseteq U^\infty$.

  (1) $P \in \Delta^0_2$.

  (2) There is a computable function $g : U^\infty \to \{0, 1\}$
  defined on $\{(n \,.\, \vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$
  such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s \,.\, \vec{x})$.

Lemma (First Part of Limit Lemma)

If (1), then (2).

Lemma (Second Part of Limit Lemma)

If (2), then (1).

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_{\mathcal{A}}\}$

- Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.
- χ_P – the characteristic function of the problem P.
- Let $P \subseteq U^\infty$.
  (1) $P \in \Delta_2^0$.
  (2) There is a computable function $g : U^\infty \to \{0, 1\}$
  defined on $\{(n \,.\, \vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$
  such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s \,.\, \vec{x})$.

Lemma (First Part of Limit Lemma)

If (1), then (2).

Lemma (Second Part of Limit Lemma)

If (2), then (1).

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_{\mathcal{A}}\}$

- Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.
- χ_P – the characteristic function of the problem P.
- Let $P \subseteq U^\infty$.

  (1) $P \in \Delta_2^0$.

  (2) There is a computable function $g : U^\infty \to \{0, 1\}$
  defined on $\{(n \,.\, \vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$
  such that $\chi_P(\vec{x}) = \lim_{s \to \infty} g(s \,.\, \vec{x})$.

Lemma (First Part of Limit Lemma)

If (1), then (2).

Lemma (Second Part of Limit Lemma)

If (2), then (1).

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

Lemma (First Part of Limit Lemma)

If $P \subseteq U^\infty$ is in Δ_2^0, then there is a computable function $g : U^\infty \to \{0, 1\}$ defined on $\{(n \,.\, \vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$ such that $\chi_P(\vec{x}) = \lim_{s \to \infty} g(s \,.\, \vec{x})$.

Proof. Let $\mathcal{M} \in \mathsf{M}_\mathcal{A}(\mathbb{H}_\mathcal{A})$ decide the problem P and $\vec{x} \in U^\infty$.

- let $\beta_1, \beta_2, \ldots, \beta_k \in \mathbb{N}$ represent the answers of the queries $(\vec{y}^{(i)} \,.\, c_{\mathcal{L}_i}) \in \mathbb{H}_\mathcal{A}$? executed by $\mathcal{M}$ on input $\vec{x}$.
- $\Rightarrow$
- $\beta_i = 0$ iff $\mathcal{L}_i(\vec{y}^{(i)}) \uparrow$,
- $\beta_i = t > 0$ iff $\mathcal{L}_i(\vec{y}^{(i)}) \uparrow^{t-1}$ and $\mathcal{L}_i(\vec{y}^{(i)}) \downarrow^{t}$.

A Characterization of $\Delta^0_2 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

Lemma (First Part of Limit Lemma)

If $P \subseteq U^\infty$ is in Δ^0_2, then there is a computable function $g : U^\infty \to \{0,1\}$ defined on $\{(n \,.\, \vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$ such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s \,.\, \vec{x})$.

Proof. Let $\mathcal{M} \in \mathsf{M}_\mathcal{A}(\mathbb{H}_\mathcal{A})$ decide the problem P and $\vec{x} \in U^\infty$.

- let $\beta_1, \beta_2, \ldots, \beta_k \in \mathbb{N}$ represent the answers of the queries $(\vec{y}^{(i)} \,.\, c_{\mathcal{L}_i}) \in \mathbb{H}_\mathcal{A}$? executed by $\mathcal{M}$ on input $\vec{x}$.
- $\Rightarrow$
- $\beta_i = 0 \quad$ iff $\mathcal{L}_i(\vec{y}^{(i)}) \uparrow$,
- $\beta_i = t > 0$ iff $\mathcal{L}_i(\vec{y}^{(i)}) \uparrow^{t-1}$ and $\mathcal{L}_i(\vec{y}^{(i)}) \downarrow^t$.

A Characterization of $\Delta^0_2 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

Lemma (First Part of Limit Lemma)

If $P \subseteq U^\infty$ is in Δ^0_2, then there is a computable function $g : U^\infty \to \{0,1\}$ defined on $\{(n\,.\,\vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$ such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s\,.\,\vec{x})$.

Proof. Let $\mathcal{M} \in \mathsf{M}_\mathcal{A}(\mathbb{H}_\mathcal{A})$ decide the problem P and $\vec{x} \in U^\infty$.

- let $\beta_1, \beta_2, \ldots, \beta_k \in \mathbb{N}$ represent the answers of the queries $(\vec{y}^{(i)}\,.\,c_{\mathcal{L}_i}) \in \mathbb{H}_\mathcal{A}$? executed by $\mathcal{M}$ on input $\vec{x}$.
- $\Rightarrow$
- $\beta_i = 0$ iff $\mathcal{L}_i(\vec{y}^{(i)}) \uparrow$,
- $\beta_i = t > 0$ iff $\mathcal{L}_i(\vec{y}^{(i)}) \uparrow^{t-1}$ and $\mathcal{L}_i(\vec{y}^{(i)}) \downarrow^{t}$.

A Characterization of $\Delta^0_2 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

Lemma (First Part of Limit Lemma)

If $P \subseteq U^\infty$ is in Δ^0_2, then there is a computable function $g : U^\infty \to \{0,1\}$ defined on $\{(n\,.\,\vec{x}) \mid n \in \mathbb{N}\ \&\ \vec{x} \in U^\infty\}$ such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s\,.\,\vec{x})$.

Proof. Let $\mathcal{M} \in \mathrm{M}_\mathcal{A}(\mathbb{H}_\mathcal{A})$ decide the problem P and $\vec{x} \in U^\infty$.

- let $\beta_1, \beta_2, \ldots, \beta_k \in \mathbb{N}$ represent the answers of the queries $(\vec{y}^{(i)}\,.\,c_{\mathcal{L}_i}) \in \mathbb{H}_\mathcal{A}$? executed by $\mathcal{M}$ on input $\vec{x}$.
- $\Rightarrow$
- $\beta_i = 0$ iff $\mathcal{L}_i(\vec{y}^{(i)}) \uparrow$,
- $\beta_i = t > 0$ iff $\mathcal{L}_i(\vec{y}^{(i)}) \uparrow^{t-1}$ and $\mathcal{L}_i(\vec{y}^{(i)}) \downarrow^t$.

A Characterization of $\Delta^0_2 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

Lemma (First Part of Limit Lemma)

If $P \subseteq U^\infty$ is in Δ^0_2, then there is a computable function $g : U^\infty \to \{0,1\}$ defined on $\{(n\,.\,\vec{x}) \mid n \in \mathbb{N}\ \&\ \vec{x} \in U^\infty\}$ such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s\,.\,\vec{x})$.

Proof. Let $\mathcal{M} \in \mathsf{M}_\mathcal{A}(\mathbb{H}_\mathcal{A})$ decide the problem P and $\vec{x} \in U^\infty$.

- let $\beta_1, \beta_2, \ldots, \beta_k \in \mathbb{N}$ represent the answers of the queries $(\vec{y}^{(i)}\,.\,c_{\mathcal{L}_i}) \in \mathbb{H}_\mathcal{A}?$ executed by $\mathcal{M}$ on input $\vec{x}$.
- $\Rightarrow$
- $\beta_i = 0$ iff $\mathcal{L}_i(\vec{y}^{(i)}) \uparrow$,
- $\beta_i = t > 0$ iff $\mathcal{L}_i(\vec{y}^{(i)}) \uparrow^{t-1}$ and $\mathcal{L}_i(\vec{y}^{(i)}) \downarrow^{t}$.

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

- Let g be computed by $\mathcal{N} \in \mathsf{M}_\mathcal{A}$ executing:
  - Input $(s \,.\, \vec{x}) \in U^\infty$;
  - if $s \in \mathbb{N}$, then simulate s instructions of $\mathcal{M}$, but, instead of

    $$\textit{if } (\vec{y}^{(s,i)} \,.\, c_{\mathcal{L}_{s,i}}) \in \mathbb{H}_\mathcal{A} \textit{ then goto to } l_1 \textit{ else goto } l_2,$$

    use

    $$\textit{if } \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \downarrow^s \textit{ then goto to } l_1 \textit{ else goto } l_2.$$
  - If the output of $\mathcal{M}$ is not reached within s steps of $\mathcal{M}$, then output 0.
- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ let $\beta_1^{(s)}, \ldots, \beta_{m_s}^{(s)} \leq s$ ($m_s \leq s$) with

  $$\beta_i^{(s)} \neq 0 \Rightarrow \; \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \uparrow^{\beta_i^{(s)}-1} \text{ and } \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \downarrow^{\beta_i^{(s)}}.$$
- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ there are $0 = s_0 < s_1 \leq s_2 \leq \cdots \leq s_k$ such that

  $$\begin{array}{rcll} (\beta_1, \beta_2, \ldots, \beta_i) & = & (\beta_1^{s_i}, \beta_2^{s_i}, \ldots, \beta_i^{s_i}) & \text{for } i \leq m_{s_i}, \\ (\beta_1, \beta_2, \ldots, \beta_k) & = & (\beta_1^{s}, \beta_2^{s}, \ldots, \beta_k^{s}) & \text{for } s \geq s_k. \end{array}$$
- $\vec{x} \in U^\infty \Rightarrow$ there is an $s_{\vec{x}}$ such that $\mathcal{N}$ outputs the same value as $\mathcal{M}$ on $(s \,.\, \vec{x})$ for all $s \geq s_{\vec{x}}$.

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

- Let g be computed by $\mathcal{N} \in \mathsf{M}_\mathcal{A}$ executing:
  - Input $(s \,.\, \vec{x}) \in U^\infty$;
  - if $s \in \mathbb{N}$, then simulate s instructions of $\mathcal{M}$, but, instead of

    $$\textit{if } (\vec{y}^{(s,i)} \,.\, c_{\mathcal{L}_{s,i}}) \in \mathbb{H}_\mathcal{A} \textit{ then goto to } l_1 \textit{ else goto } l_2,$$

    use

    $$\textit{if } \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \downarrow^s \textit{ then goto to } l_1 \textit{ else goto } l_2.$$
  - If the output of $\mathcal{M}$ is not reached within s steps of $\mathcal{M}$, then output 0.
- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ let $\beta_1^{(s)}, \ldots, \beta_{m_s}^{(s)} \leq s$ $(m_s \leq s)$ with

  $$\beta_i^{(s)} \neq 0 \Rightarrow \ \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \uparrow^{\beta_i^{(s)}-1} \text{ and } \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \downarrow^{\beta_i^{(s)}} .$$
- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ there are $0 = s_0 < s_1 \leq s_2 \leq \cdots \leq s_k$ such that

  $$\begin{aligned} (\beta_1, \beta_2, \ldots, \beta_i) &= (\beta_1^{s_i}, \beta_2^{s_i}, \ldots, \beta_i^{s_i}) && \text{for } i \leq m_{s_i}, \\ (\beta_1, \beta_2, \ldots, \beta_k) &= (\beta_1^{s}, \beta_2^{s}, \ldots, \beta_k^{s}) && \text{for } s \geq s_k. \end{aligned}$$
- $\vec{x} \in U^\infty \Rightarrow$ there is an $s_{\vec{x}}$ such that $\mathcal{N}$ outputs the same value as $\mathcal{M}$ on $(s \,.\, \vec{x})$ for all $s \geq s_{\vec{x}}$.

A Characterization of $\Delta^0_2 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

- Let g be computed by $\mathcal{N} \in \mathsf{M}_\mathcal{A}$ executing:
  - Input $(s \,.\, \vec{x}) \in U^\infty$;
  - if $s \in \mathbb{N}$, then simulate s instructions of $\mathcal{M}$, but, instead of

$$\textit{if } (\vec{y}^{(s,i)} \,.\, c_{\mathcal{L}_{s,i}}) \in \mathbb{H}_\mathcal{A} \textit{ then goto to } l_1 \textit{ else goto } l_2,$$

use

$$\textit{if } \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \downarrow^s \textit{ then goto to } l_1 \textit{ else goto } l_2.$$

  - If the output of $\mathcal{M}$ is not reached within s steps of $\mathcal{M}$, then output 0.
- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ let $\beta_1^{(s)}, \ldots, \beta_{m_s}^{(s)} \leq s$ ($m_s \leq s$) with

$$\beta_i^{(s)} \neq 0 \Rightarrow \; \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \uparrow^{\beta_i^{(s)}-1} \text{ and } \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \downarrow^{\beta_i^{(s)}} .$$

- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ there are $0 = s_0 < s_1 \leq s_2 \leq \cdots \leq s_k$ such that

$$\begin{aligned} (\beta_1, \beta_2, \ldots, \beta_i) &= (\beta_1^{s_i}, \beta_2^{s_i}, \ldots, \beta_i^{s_i}) && \text{for } i \leq m_{s_i}, \\ (\beta_1, \beta_2, \ldots, \beta_k) &= (\beta_1^{s}, \beta_2^{s}, \ldots, \beta_k^{s}) && \text{for } s \geq s_k. \end{aligned}$$

- $\vec{x} \in U^\infty \Rightarrow$ there is an $s_{\vec{x}}$ such that $\mathcal{N}$ outputs the same value as $\mathcal{M}$ on $(s \,.\, \vec{x})$ for all $s \geq s_{\vec{x}}$.

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

- Let g be computed by $\mathcal{N} \in \mathsf{M}_\mathcal{A}$ executing:
  - Input $(s\,.\,\vec{x}) \in U^\infty$;
  - if $s \in \mathbb{N}$, then simulate s instructions of $\mathcal{M}$, but, instead of

$$\textit{if } (\vec{y}^{(s,i)}\,.\,c_{\mathcal{L}_{s,i}}) \in \mathbb{H}_\mathcal{A} \textit{ then goto to } l_1 \textit{ else goto } l_2,$$

    use

$$\textit{if } \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \downarrow^s \textit{ then goto to } l_1 \textit{ else goto } l_2.$$

  - If the output of $\mathcal{M}$ is not reached within s steps of $\mathcal{M}$, then output 0.
- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ let $\beta_1^{(s)}, \ldots, \beta_{m_s}^{(s)} \leq s$ ($m_s \leq s$) with

$$\beta_i^{(s)} \neq 0 \Rightarrow \; \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \uparrow^{\beta_i^{(s)}-1} \text{ and } \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \downarrow^{\beta_i^{(s)}} .$$

- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ there are $0 = s_0 < s_1 \leq s_2 \leq \cdots \leq s_k$ such that

$$\begin{array}{rcll} (\beta_1, \beta_2, \ldots, \beta_i) & = & (\beta_1^{s_i}, \beta_2^{s_i}, \ldots, \beta_i^{s_i}) & \text{for } i \leq m_{s_i}, \\ (\beta_1, \beta_2, \ldots, \beta_k) & = & (\beta_1^{s}, \beta_2^{s}, \ldots, \beta_k^{s}) & \text{for } s \geq s_k. \end{array}$$

- $\vec{x} \in U^\infty \Rightarrow$ there is an $s_{\vec{x}}$ such that $\mathcal{N}$ outputs the same value as $\mathcal{M}$ on $(s\,.\,\vec{x})$ for all $s \geq s_{\vec{x}}$.

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_{\mathcal{A}}\}$

- Let g be computed by $\mathcal{N} \in \mathsf{M}_{\mathcal{A}}$ executing:
  - Input $(s \,.\, \vec{x}) \in U^\infty$;
  - if $s \in \mathbb{N}$, then simulate s instructions of $\mathcal{M}$, but, instead of

    $$\textit{if } (\vec{y}^{(s,i)} \,.\, c_{\mathcal{L}_{s,i}}) \in \mathbb{H}_{\mathcal{A}} \textit{ then goto to } l_1 \textit{ else goto } l_2,$$

    use

    $$\textit{if } \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \downarrow^s \textit{ then goto to } l_1 \textit{ else goto } l_2.$$
  - If the output of $\mathcal{M}$ is not reached within s steps of $\mathcal{M}$, then output 0.
- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ let $\beta_1^{(s)}, \ldots, \beta_{m_s}^{(s)} \leq s$ ($m_s \leq s$) with

  $$\beta_i^{(s)} \neq 0 \Rightarrow \ \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \uparrow^{\beta_i^{(s)}-1} \ \text{ and } \mathcal{L}_{s,i}(\vec{y}^{(s,i)}) \downarrow^{\beta_i^{(s)}} \ .$$
- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ there are $0 = s_0 < s_1 \leq s_2 \leq \cdots \leq s_k$ such that

  $$\begin{array}{rcll} (\beta_1, \beta_2, \ldots, \beta_i) & = & (\beta_1^{s_i}, \beta_2^{s_i}, \ldots, \beta_i^{s_i}) & \text{for } i \leq m_{s_i}, \\ (\beta_1, \beta_2, \ldots, \beta_k) & = & (\beta_1^{s}, \beta_2^{s}, \ldots, \beta_k^{s}) & \text{for } s \geq s_k. \end{array}$$
- $\vec{x} \in U^\infty \Rightarrow$ there is an $s_{\vec{x}}$ such that $\mathcal{N}$ outputs the same value as $\mathcal{M}$ on $(s \,.\, \vec{x})$ for all $s \geq s_{\vec{x}}$.

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_{\mathcal{A}}\}$

Lemma (Second Part of Limit Lemma)

If there is a computable function $g : U^\infty \to \{0,1\}$ defined on $\{(n \,.\, \vec{x}) \mid n \in \mathbb{N} \,\&\, \vec{x} \in U^\infty\}$ such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s \,.\, \vec{x})$, then $P \subseteq U^\infty$ is in Δ_2^0.

Proof. Let g be computed by $\mathcal{N} \in \mathsf{M}_{\mathcal{A}}$ and let $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}(\mathbb{H}_{\mathcal{A}})$ execute:

- Input $\vec{x} \in U^\infty$;
- Let $s = 1$;
- 1:

  \- Ask $((s \,.\, \vec{x}) \,.\, c_{\mathcal{L}}) \in \mathbb{H}_{\mathcal{A}}$? where
  $\mathcal{L}$: Input $(s \,.\, \vec{x})$;
  Halt if there is a $k \geq s$ such that $g(s \,.\, \vec{x}) \neq g(k \,.\, \vec{x})$.
  If $\mathcal{L}(s \,.\, \vec{x}) \downarrow$,
  then $s := s + 1$ and goto 1
  else compute $g(s \,.\, \vec{x})$ by simulating $\mathcal{N}$ and output $g(s \,.\, \vec{x})$.

$\mathcal{M}$ decides P.

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_{\mathcal{A}}\}$

Lemma (Second Part of Limit Lemma)

If there is a computable function $g : U^\infty \to \{0,1\}$ defined on $\{(n\,.\,\vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$ such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s\,.\,\vec{x})$, then $P \subseteq U^\infty$ is in Δ_2^0.

Proof. Let g be computed by $\mathcal{N} \in \mathsf{M}_{\mathcal{A}}$ and let $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}(\mathbb{H}_{\mathcal{A}})$ execute:

- Input $\vec{x} \in U^\infty$;
- Let $s = 1$;
- 1:

  - Ask $((s\,.\,\vec{x})\,.\,c_{\mathcal{L}}) \in \mathbb{H}_{\mathcal{A}}$? where

    $\mathcal{L}$: Input $(s\,.\,\vec{x})$;

    Halt if there is a $k \geq s$ such that $g(s\,.\,\vec{x}) \neq g(k\,.\,\vec{x})$.

  If $\mathcal{L}(s\,.\,\vec{x}) \downarrow$,

  then $s := s + 1$ and goto 1

  else compute $g(s\,.\,\vec{x})$ by simulating $\mathcal{N}$ and output $g(s\,.\,\vec{x})$.

$\mathcal{M}$ decides P.

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_{\mathcal{A}}\}$

Lemma (Second Part of Limit Lemma)

If there is a computable function $g : U^\infty \to \{0,1\}$ defined on $\{(n \,.\, \vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$ such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s \,.\, \vec{x})$, then $P \subseteq U^\infty$ is in Δ_2^0.

Proof. Let g be computed by $\mathcal{N} \in \mathsf{M}_{\mathcal{A}}$ and let $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}(\mathbb{H}_{\mathcal{A}})$ execute:

- Input $\vec{x} \in U^\infty$;
- Let $s = 1$;
- 1:

  - Ask $((s \,.\, \vec{x}) \,.\, c_{\mathcal{L}}) \in \mathbb{H}_{\mathcal{A}}$? where

    $\mathcal{L}$: Input $(s \,.\, \vec{x})$;

    Halt if there is a $k \geq s$ such that $g(s \,.\, \vec{x}) \neq g(k \,.\, \vec{x})$.

  If $\mathcal{L}(s \,.\, \vec{x}) \downarrow$,

  then $s := s + 1$ and goto 1

  else compute $g(s \,.\, \vec{x})$ by simulating $\mathcal{N}$ and output $g(s \,.\, \vec{x})$.

$\mathcal{M}$ decides P.

A Characterization of $\Delta_2^0 = \{P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_\mathcal{A}\}$

Lemma (Second Part of Limit Lemma)

If there is a computable function $g : U^\infty \to \{0,1\}$ defined on $\{(n \,.\, \vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$ such that $\chi_P(\vec{x}) = \lim_{s\to\infty} g(s \,.\, \vec{x})$, then $P \subseteq U^\infty$ is in Δ_2^0.

Proof. Let g be computed by $\mathcal{N} \in \mathsf{M}_\mathcal{A}$ and let $\mathcal{M} \in \mathsf{M}_\mathcal{A}(\mathbb{H}_\mathcal{A})$ execute:

- Input $\vec{x} \in U^\infty$;
- Let $s = 1$;
- 1:

  \- Ask $((s \,.\, \vec{x}) \,.\, c_\mathcal{L}) \in \mathbb{H}_\mathcal{A}$? where

  $\mathcal{L}$: Input $(s \,.\, \vec{x})$;

  Halt if there is a $k \geq s$ such that $g(s \,.\, \vec{x}) \neq g(k \,.\, \vec{x})$.

  If $\mathcal{L}(s \,.\, \vec{x}) \downarrow$,

  then $s := s + 1$ and goto 1

  else compute $g(s \,.\, \vec{x})$ by simulating $\mathcal{N}$ and output $g(s \,.\, \vec{x})$.

$\mathcal{M}$ decides P.

Summary: Δ_2^0 and the Limit Lemma

- Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.
- χ_P – the characteristic function of the problem P.

Lemma (Limit Lemma)

$P \subseteq U^\infty$ is in Δ_2^0 if and only if there is a computable function $g : U^\infty \to \{0, 1\}$ defined on $\{(n \,.\, \vec{x}) \mid n \in \mathbb{N} \;\&\; \vec{x} \in U^\infty\}$ such that $\chi_P(\vec{x}) = \lim_{s \to \infty} g(s \,.\, \vec{x})$.

- Let $\mathcal{A}$ contain
  - only a finite number of operations and relations,
  - an effectively enumerable set denoted by $\mathbb{N}$,
  - only two constants denoted by 0 and 1.
- $\Rightarrow \mathbb{H}_\mathcal{A} \in \mathbf{REC}_\mathcal{A}$.
- We construct an $\mathbb{A} \subset \mathbb{N}$ with
  - $\mathbb{A} \not\precsim_T \mathbb{H}_\mathcal{A}$,
  - $\mathbb{A} \not\leq_T P$ for many $P \subseteq U^\infty$,
  - $\mathbb{A} \not\leq_T P$ for many $P \in \Delta_2^0$.
  - $\Rightarrow \mathbb{H}_\mathcal{A} \not\leq_T P$.

A Generalization of the Friedberg-Muchnik Theorem

- Let $\mathcal{A}$ contain
  - only a finite number of operations and relations,
  - an effectively enumerable set denoted by $\mathbb{N}$,
  - only two constants denoted by 0 and 1.
- $\Rightarrow \mathbb{H}_{\mathcal{A}} \in \mathrm{REC}_{\mathcal{A}}$.
- We construct an $\mathbb{A} \subset \mathbb{N}$ with
  - $\mathbb{A} \not\preceq_T \mathbb{H}_{\mathcal{A}}$,
  - $\mathbb{A} \not\succeq_T P$ for many $P \subseteq U^{\infty}$,
  - $\mathbb{A} \not\succeq_T P$ for many $P \in \Delta_2^0$.
  - $\Rightarrow \mathbb{H}_{\mathcal{A}} \not\succeq_T P$.

A Generalization of the Friedberg-Muchnik Theorem

- Let $\mathcal{A}$ contain
  - only a finite number of operations and relations,
  - an effectively enumerable set denoted by $\mathbb{N}$,
  - only two constants denoted by 0 and 1.
- $\Rightarrow \mathbb{H}_{\mathcal{A}} \in \mathrm{REC}_{\mathcal{A}}$.
- We construct an $\mathbb{A} \subset \mathbb{N}$ with
  - $\mathbb{A} \precneqq_T \mathbb{H}_{\mathcal{A}}$,
  - $\mathbb{A} \not\preceq_T P$ for many $P \subseteq U^{\infty}$,
  - $\mathbb{A} \not\preceq_T P$ for many $P \in \Delta_2^0$.
  - $\Rightarrow \mathbb{H}_{\mathcal{A}} \not\preceq_T P$.

A Generalization of the Friedberg-Muchnik Theorem

$\mathcal{A}$ contains

- only a finite number of operations and relations,
- an effectively enumerable set denoted by $\mathbb{N}$,
- only two constants denoted by 0 and 1.

We use:

- $H_{\mathcal{M}} \cap \mathbb{N}$ (for the halting set $H_{\mathcal{M}}$ of $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}$) is
  - effectively enumerable
  - a halting set of a machine in $\mathsf{M}_{\mathcal{A}}$.
- For any $\mathcal{O} \subseteq \mathbb{N}$,
  we can list $\mathsf{M}_{\mathcal{A}}(\mathcal{O})$: $\mathcal{M}_1^{\mathcal{O}}, \mathcal{M}_2^{\mathcal{O}}, \ldots$.
  (The index is the code of the corresponding program.)
- We can list $\mathsf{M}_{\mathcal{A}}$: $\mathcal{N}_1, \mathcal{N}_2, \ldots$.
- $\bar{\mathcal{N}}_i$ enumerating all positive integers $n_{i,1}, n_{i,2}, \ldots \in H_{\mathcal{N}_i}$.

A Generalization of the Friedberg-Muchnik Theorem

$\mathcal{A}$ contains

- only a finite number of operations and relations,
- an effectively enumerable set denoted by $\mathbb{N}$,
- only two constants denoted by 0 and 1.

We use:

- $H_{\mathcal{M}} \cap \mathbb{N}$ (for the halting set $H_{\mathcal{M}}$ of $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}$) is
  - effectively enumerable
  - a halting set of a machine in $\mathsf{M}_{\mathcal{A}}$.
- For any $\mathcal{O} \subseteq \mathbb{N}$, we can list $\mathsf{M}_{\mathcal{A}}(\mathcal{O})$: $\mathcal{M}_1^{\mathcal{O}}, \mathcal{M}_2^{\mathcal{O}}, \ldots$. (The index is the code of the corresponding program.)
- We can list $\mathsf{M}_{\mathcal{A}}$: $\mathcal{N}_1, \mathcal{N}_2, \ldots$.
- $\bar{\mathcal{N}}_i$ enumerating all positive integers $n_{i,1}, n_{i,2}, \ldots \in H_{\mathcal{N}_i}$.

A Generalization of the Friedberg-Muchnik Theorem

$\mathcal{A}$ contains

- only a finite number of operations and relations,
- an effectively enumerable set denoted by $\mathbb{N}$,
- only two constants denoted by 0 and 1.

We use:

- $H_{\mathcal{M}} \cap \mathbb{N}$ (for the halting set $H_{\mathcal{M}}$ of $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}$) is
  - effectively enumerable
  - a halting set of a machine in $\mathsf{M}_{\mathcal{A}}$.
- For any $\mathcal{O} \subseteq \mathbb{N}$,
  we can list $\mathsf{M}_{\mathcal{A}}(\mathcal{O})$: $\mathcal{M}_1^{\mathcal{O}}, \mathcal{M}_2^{\mathcal{O}}, \ldots$.
  (The index is the code of the corresponding program.)
- We can list $\mathsf{M}_{\mathcal{A}}$: $\mathcal{N}_1, \mathcal{N}_2, \ldots$.
- $\bar{\mathcal{N}}_i$ enumerating all positive integers $n_{i,1}, n_{i,2}, \ldots \in H_{\mathcal{N}_i}$.

$\mathcal{A}$ contains

- only a finite number of operations and relations,
- an effectively enumerable set denoted by $\mathbb{N}$,
- only two constants denoted by 0 and 1.

We use:

- $H_{\mathcal{M}} \cap \mathbb{N}$ (for the halting set $H_{\mathcal{M}}$ of $\mathcal{M} \in$ M$_{\mathcal{A}}$) is
  - effectively enumerable
  - a halting set of a machine in M$_{\mathcal{A}}$.
- For any $\mathcal{O} \subseteq \mathbb{N}$,
  we can list M$_{\mathcal{A}}(\mathcal{O})$: $\mathcal{M}_1^{\mathcal{O}}, \mathcal{M}_2^{\mathcal{O}}, \ldots$.
  (The index is the code of the corresponding program.)
- We can list M$_{\mathcal{A}}$: $\mathcal{N}_1, \mathcal{N}_2, \ldots$.
- $\bar{\mathcal{N}}_i$ enumerating all positive integers $n_{i,1}, n_{i,2}, \ldots \in H_{\mathcal{N}_i}$.

A Generalization of the Friedberg-Muchnik Theorem

$\mathcal{A}$ contains

- only a finite number of operations and relations,
- an effectively enumerable set denoted by $\mathbb{N}$,
- only two constants denoted by 0 and 1.

We use:

- $H_{\mathcal{M}} \cap \mathbb{N}$ (for the halting set $H_{\mathcal{M}}$ of $\mathcal{M} \in \mathrm{M}_{\mathcal{A}}$) is
  - effectively enumerable
  - a halting set of a machine in $\mathrm{M}_{\mathcal{A}}$.
- For any $\mathcal{O} \subseteq \mathbb{N}$,
  we can list $\mathrm{M}_{\mathcal{A}}(\mathcal{O})$: $\mathcal{M}_1^{\mathcal{O}}, \mathcal{M}_2^{\mathcal{O}}, \ldots$.
  (The index is the code of the corresponding program.)
- We can list $\mathrm{M}_{\mathcal{A}}$: $\mathcal{N}_1, \mathcal{N}_2, \ldots$.
- $\bar{\mathcal{N}}_i$ enumerating all positive integers $n_{i,1}, n_{i,2}, \ldots \in H_{\mathcal{N}_i}$.

A Generalization of the Friedberg-Muchnik Theorem

Let $\mathbb{A} = \bigcup_{s \geq 0} \mathbb{A}_s$ be defined in stages. $\mathbb{A}_0 = \emptyset$, $s \geq 0$.

$$I_s = \{i \leq s \mid W_{i,s} \cap \mathbb{A}_s = \emptyset \ \& \ (\exists x \in W_{i,s})(2i < x \ \& \ (\forall j \leq i)(a(j,s) < x))\}$$

where, for any $j \leq s$,

- $a(j,s) \begin{cases} \text{greatest integer used in a query by} & \\ \mathcal{M}_j^{\mathbb{A}_s} \text{ on } j \text{ within } s \text{ steps} & \text{if } \mathcal{M}_j^{\mathbb{A}_s}(j) \downarrow^s, \\ 0 & \text{if } \mathcal{M}_j^{\mathbb{A}_s}(j) \uparrow^s. \end{cases}$
- $W_{i,s}$ is the set of integers computed by $\bar{\mathcal{N}}_i$ on s within s steps.

If $I_s \neq \emptyset$, then let

$$\begin{aligned} i_s &= \min I_s, \\ x_{i_s} &= \min\{x \in W_{i_s,s} \mid 2i_s < x \ \& \ (\forall j \leq i_s)(a(j,s) < x)\}, \\ \mathbb{A}_{s+1} &= \begin{cases} \mathbb{A}_s & \text{if } I_s = \emptyset \\ \mathbb{A}_s \cup \{x_{i_s}\} & \text{otherwise.} \end{cases} \end{aligned}$$

A Generalization of the Friedberg-Muchnik Theorem

Let $\mathbb{A} = \bigcup_{s \geq 0} \mathbb{A}_s$ be defined in stages. $\mathbb{A}_0 = \emptyset$, $s \geq 0$.

$$I_s = \{i \leq s \mid W_{i,s} \cap \mathbb{A}_s = \emptyset \;\&\; (\exists x \in W_{i,s})(2i < x \;\&\; (\forall j \leq i)(a(j,s) < x))\}$$

where, for any $j \leq s$,

- $a(j,s) \begin{cases} \text{greatest integer used in a query by} & \\ \mathcal{M}_j^{\mathbb{A}_s} \text{ on } j \text{ within } s \text{ steps} & \text{if } \mathcal{M}_j^{\mathbb{A}_s}(j)\downarrow^s, \\ 0 & \text{if } \mathcal{M}_j^{\mathbb{A}_s}(j)\uparrow^s. \end{cases}$
- $W_{i,s}$ is the set of integers computed by $\bar{\mathcal{N}}_i$ on s within s steps.

If $I_s \neq \emptyset$, then let

$$\begin{aligned} i_s &= \min I_s, \\ x_{i_s} &= \min\{x \in W_{i_s,s} \mid 2i_s < x \;\&\; (\forall j \leq i_s)(a(j,s) < x)\}, \\ \mathbb{A}_{s+1} &= \begin{cases} \mathbb{A}_s & \text{if } I_s = \emptyset \\ \mathbb{A}_s \cup \{x_{i_s}\} & \text{otherwise.} \end{cases} \end{aligned}$$

A Generalization of the Friedberg-Muchnik Theorem

The properties of $\mathbb{A}$, for instance, for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdots; \leq)$:

- $\mathbb{A}$ is effectively enumerable by an machine in $\mathsf{M}_{\mathcal{A}}$.

  $\Rightarrow \mathbb{A} \preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{A}$ and $\mathbb{N} \setminus \mathbb{A}$ are infinite.

- Conditions for lowness for all $n > 0$:
  (N_n) If $\mathcal{M}_n^{\mathbb{A}_t}(n) \downarrow^t$ for infinitely many t, then $\mathcal{M}_n^{\mathbb{A}}(n) \downarrow$.

- $\Rightarrow$ Conditions for simplicity for all $n > 0$:
  (P_n) If $W_n = \bigcup_{i \geq 1} W_{n,i}$ is infinite, then $\mathbb{A} \cap W_n \neq \emptyset$.

  $\Rightarrow \mathbb{A}^C \not\preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{K}^{\mathbb{A}} \preceq_T \mathbb{K}^{\emptyset}$ where $\mathbb{K}^{\mathcal{O}} = \{k_{\mathcal{M}} \mid \mathcal{M} \in \mathsf{M}^1_{\mathcal{A}}(\mathcal{O})\ \&\ \mathcal{M}(k_{\mathcal{M}}) \downarrow\}$.

  $\Rightarrow \mathbb{A} \not\preceq_T \mathbb{H}_{\mathcal{A}}$.

A Generalization of the Friedberg-Muchnik Theorem

The properties of $\mathbb{A}$, for instance, for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdots; \leq)$:

- $\mathbb{A}$ is effectively enumerable by an machine in $\mathsf{M}_{\mathcal{A}}$.

  $\Rightarrow \mathbb{A} \preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{A}$ and $\mathbb{N} \setminus \mathbb{A}$ are infinite.
- Conditions for lowness for all $n > 0$:
  (N_n) If $\mathcal{M}_n^{\mathbb{A}_t}(n) \downarrow^t$ for infinitely many t, then $\mathcal{M}_n^{\mathbb{A}}(n) \downarrow$.
- $\Rightarrow$ Conditions for simplicity for all $n > 0$:
  (P_n) If $W_n = \bigcup_{i \geq 1} W_{n,i}$ is infinite, then $\mathbb{A} \cap W_n \neq \emptyset$.

  $\Rightarrow \mathbb{A}^C \not\preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{K}^{\mathbb{A}} \preceq_T \mathbb{K}^{\emptyset}$ where $\mathbb{K}^{\mathcal{O}} = \{k_{\mathcal{M}} \mid \mathcal{M} \in \mathsf{M}_{\mathcal{A}}^1(\mathcal{O})\ \&\ \mathcal{M}(k_{\mathcal{M}}) \downarrow\}$.

  $\Rightarrow \mathbb{A} \npreceq_T \mathbb{H}_{\mathcal{A}}$.

A Generalization of the Friedberg-Muchnik Theorem

The properties of $\mathbb{A}$, for instance, for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdots; \leq)$:

- $\mathbb{A}$ is effectively enumerable by an machine in $\mathsf{M}_{\mathcal{A}}$.

  $\Rightarrow \mathbb{A} \preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{A}$ and $\mathbb{N} \setminus \mathbb{A}$ are infinite.
- Conditions for lowness for all $n > 0$:
  (N_n) If $\mathcal{M}_n^{\mathbb{A}_t}(n) \downarrow^t$ for infinitely many t, then $\mathcal{M}_n^{\mathbb{A}}(n) \downarrow$.
- $\Rightarrow$ Conditions for simplicity for all $n > 0$:
  (P_n) If $W_n = \bigcup_{i \geq 1} W_{n,i}$ is infinite, then $\mathbb{A} \cap W_n \neq \emptyset$.

  $\Rightarrow \mathbb{A}^C \not\preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{K}^{\mathbb{A}} \preceq_T \mathbb{K}^{\emptyset}$ where $\mathbb{K}^{\mathcal{O}} = \{k_{\mathcal{M}} \mid \mathcal{M} \in \mathsf{M}^1_{\mathcal{A}}(\mathcal{O})\ \&\ \mathcal{M}(k_{\mathcal{M}}) \downarrow\}$.

  $\Rightarrow \mathbb{A} \npreceq_T \mathbb{H}_{\mathcal{A}}$.

A Generalization of the Friedberg-Muchnik Theorem

The properties of $\mathbb{A}$, for instance, for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdots; \leq)$:

- $\mathbb{A}$ is effectively enumerable by an machine in $\mathsf{M}_{\mathcal{A}}$.

  $\Rightarrow \mathbb{A} \preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{A}$ and $\mathbb{N} \setminus \mathbb{A}$ are infinite.

- Conditions for lowness for all $n > 0$:
  (N_n) If $\mathcal{M}_n^{\mathbb{A}_t}(n) \downarrow^t$ for infinitely many t, then $\mathcal{M}_n^{\mathbb{A}}(n) \downarrow$.

- $\Rightarrow$ Conditions for simplicity for all $n > 0$:
  (P_n) If $W_n = \bigcup_{i \geq 1} W_{n,i}$ is infinite, then $\mathbb{A} \cap W_n \neq \emptyset$.

  $\Rightarrow \mathbb{A}^C \not\preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{K}^{\mathbb{A}} \preceq_T \mathbb{K}^{\emptyset}$ where $\mathbb{K}^{\mathcal{O}} = \{k_{\mathcal{M}} \mid \mathcal{M} \in \mathsf{M}^1_{\mathcal{A}}(\mathcal{O})\ \&\ \mathcal{M}(k_{\mathcal{M}}) \downarrow\}$.

  $\Rightarrow \mathbb{A} \not\succeq_T \mathbb{H}_{\mathcal{A}}$.

A Generalization of the Friedberg-Muchnik Theorem

The properties of $\mathbb{A}$, for instance, for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdots; \leq)$:

- $\mathbb{A}$ is effectively enumerable by an machine in $\mathsf{M}_{\mathcal{A}}$.

  $\Rightarrow \mathbb{A} \preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{A}$ and $\mathbb{N} \setminus \mathbb{A}$ are infinite.

- Conditions for lowness for all $n > 0$:
  (N_n) If $\mathcal{M}_n^{\mathbb{A}_t}(n) \downarrow^t$ for infinitely many t, then $\mathcal{M}_n^{\mathbb{A}}(n) \downarrow$.

- $\Rightarrow$ Conditions for simplicity for all $n > 0$:
  (P_n) If $W_n = \bigcup_{i \geq 1} W_{n,i}$ is infinite, then $\mathbb{A} \cap W_n \neq \emptyset$.

  $\Rightarrow \mathbb{A}^C \not\preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{K}^{\mathbb{A}} \preceq_T \mathbb{K}^{\emptyset}$ where $\mathbb{K}^{\mathcal{O}} = \{k_{\mathcal{M}} \mid \mathcal{M} \in \mathsf{M}^1_{\mathcal{A}}(\mathcal{O})\ \&\ \mathcal{M}(k_{\mathcal{M}}) \downarrow\}$.

  $\Rightarrow \mathbb{A} \precneqq_T \mathbb{H}_{\mathcal{A}}$.

A Generalization of the Friedberg-Muchnik Theorem

The properties of $\mathbb{A}$, for instance, for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdots; \leq)$:

- $\mathbb{A}$ is effectively enumerable by an machine in $\mathsf{M}_{\mathcal{A}}$.

  $\Rightarrow \mathbb{A} \preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{A}$ and $\mathbb{N} \setminus \mathbb{A}$ are infinite.

- Conditions for lowness for all $n > 0$:
  (N_n) If $\mathcal{M}_n^{\mathbb{A}_t}(n) \downarrow^t$ for infinitely many t, then $\mathcal{M}_n^{\mathbb{A}}(n) \downarrow$.

- $\Rightarrow$ Conditions for simplicity for all $n > 0$:
  (P_n) If $W_n = \bigcup_{i \geq 1} W_{n,i}$ is infinite, then $\mathbb{A} \cap W_n \neq \emptyset$.

  $\Rightarrow \mathbb{A}^C \not\preceq_1 \mathbb{H}_{\mathcal{A}}$.

- $\mathbb{K}^{\mathbb{A}} \preceq_T \mathbb{K}^{\emptyset}$ where $\mathbb{K}^{\mathcal{O}} = \{k_{\mathcal{M}} \mid \mathcal{M} \in \mathsf{M}^1_{\mathcal{A}}(\mathcal{O})\ \&\ \mathcal{M}(k_{\mathcal{M}}) \downarrow\}$.

  $\Rightarrow \mathbb{A} \not\preceq_T \mathbb{H}_{\mathcal{A}}$.

A Generalization of the Friedberg-Muchnik Theorem

- Let $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ or $\mathcal{A} = (\mathbb{C}; 0, 1; +, -, \cdot; =)$ and
  $P = \mathbb{A}_{\text{Alg}}$,
  $P = \bigcup_{n\geq 1}\{\vec{x} \in \mathbb{R}^n \mid (\exists \vec{q} \in \mathbb{Q}^n)(q_1 + \sum_{i=1}^{n-1} q_{i-1}x_i = x_n)\}$,
  $\mathbb{Z} \subseteq P$ or $\mathbb{Z} \cap P = \emptyset$.

Lemma

$\mathbb{A} \not\preceq_T P$.

Corollary

$\mathbb{H}_{\mathcal{A}} \not\preceq_T P$.

Remark

Similar constructions are also possible, if all problems which are semi-decidable by Turing machines are decidable over $\mathcal{A}$.

- $P \not\preceq_T \mathbb{H}_{\mathcal{A}}$ holds for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \chi_{\mathbb{H}_{\text{TM}}}, \phi; \leq)$ with $\phi(x) = \pi$ and $\pi\mathbb{Z} \subseteq P$ (where $\mathbb{A} \subseteq \pi\mathbb{Z}$) and so on.

A Generalization of the Friedberg-Muchnik Theorem

- Let $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ or $\mathcal{A} = (\mathbb{C}; 0, 1; +, -, \cdot; =)$ and
  $P = \mathbb{A}_{\mathrm{Alg}}$,
  $P = \bigcup_{n\geq 1}\{\vec{x} \in \mathbb{R}^n \mid (\exists \vec{q} \in \mathbb{Q}^n)(q_1 + \sum_{i=1}^{n-1} q_{i-1}x_i = x_n)\}$,
  $\mathbb{Z} \subseteq P$ or $\mathbb{Z} \cap P = \emptyset$.

Lemma

$\mathbb{A} \not\preceq_T P$.

Corollary

$\mathbb{H}_{\mathcal{A}} \not\preceq_T P$.

Remark

Similar constructions are also possible, if all problems which are semi-decidable by Turing machines are decidable over $\mathcal{A}$.

- $P \not\preceq_T \mathbb{H}_{\mathcal{A}}$ holds for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \chi_{\mathbb{H}_{\mathrm{TM}}}, \phi; \leq)$ with $\phi(x) = \pi$ and $\pi\mathbb{Z} \subseteq P$ (where $\mathbb{A} \subseteq \pi\mathbb{Z}$) and so on.

A Generalization of the Friedberg-Muchnik Theorem

- Let $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ or $\mathcal{A} = (\mathbb{C}; 0, 1; +, -, \cdot; =)$ and
  $P = \mathbb{A}_{\text{Alg}}$,
  $P = \bigcup_{n\geq 1}\{\vec{x} \in \mathbb{R}^n \mid (\exists \vec{q} \in \mathbb{Q}^n)(q_1 + \sum_{i=1}^{n-1} q_{i-1}x_i = x_n)\}$,
  $\mathbb{Z} \subseteq P$ or $\mathbb{Z} \cap P = \emptyset$.

Lemma

$\mathbb{A} \not\preceq_T P$.

Corollary

$\mathbb{H}_{\mathcal{A}} \not\preceq_T P$.

Remark

Similar constructions are also possible, if all problems which are semi-decidable by Turing machines are decidable over $\mathcal{A}$.

- $P \not\preceq_T \mathbb{H}_{\mathcal{A}}$ holds for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \chi_{\mathbb{H}_{\text{TM}}}, \phi; \leq)$ with $\phi(x) = \pi$ and $\pi\mathbb{Z} \subseteq P$ (where $\mathbb{A} \subseteq \pi\mathbb{Z}$) and so on.

A Generalization of the Friedberg-Muchnik Theorem

- Let $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ or $\mathcal{A} = (\mathbb{C}; 0, 1; +, -, \cdot; =)$ and
  $P = \mathbb{A}_{\text{Alg}}$,
  $P = \bigcup_{n \geq 1} \{\vec{x} \in \mathbb{R}^n \mid (\exists \vec{q} \in \mathbb{Q}^n)(q_1 + \sum_{i=1}^{n-1} q_{i-1} x_i = x_n)\}$,
  $\mathbb{Z} \subseteq P$ or $\mathbb{Z} \cap P = \emptyset$.

Lemma

$\mathbb{A} \not\preceq_T P$.

Corollary

$\mathbb{H}_{\mathcal{A}} \not\preceq_T P$.

Remark

Similar constructions are also possible, if all problems which are semi-decidable by Turing machines are decidable over $\mathcal{A}$.

- $P \not\preceq_T \mathbb{H}_{\mathcal{A}}$ holds for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \chi_{\mathbb{H}_{\text{TM}}}, \phi; \leq)$ with $\phi(x) = \pi$ and $\pi\mathbb{Z} \subseteq P$ (where $\mathbb{A} \subseteq \pi\mathbb{Z}$) and so on.

A Generalization of the Friedberg-Muchnik Theorem

- Let $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ or $\mathcal{A} = (\mathbb{C}; 0, 1; +, -, \cdot; =)$ and
  $P = \mathbb{A}_{\mathrm{Alg}}$,
  $P = \bigcup_{n \geq 1} \{\vec{x} \in \mathbb{R}^n \mid (\exists \vec{q} \in \mathbb{Q}^n)(q_1 + \sum_{i=1}^{n-1} q_{i-1} x_i = x_n)\}$,
  $\mathbb{Z} \subseteq P$ or $\mathbb{Z} \cap P = \emptyset$.

Lemma

$\mathbb{A} \not\preceq_T P$.

Corollary

$\mathbb{H}_{\mathcal{A}} \not\preceq_T P$.

Remark

Similar constructions are also possible, if all problems which are semi-decidable by Turing machines are decidable over $\mathcal{A}$.

- $P \not\preceq_T \mathbb{H}_{\mathcal{A}}$ holds for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \chi_{\mathbb{H}_{\mathrm{TM}}}, \phi; \leq)$ with $\phi(x) = \pi$ and $\pi\mathbb{Z} \subseteq P$ (where $\mathbb{A} \subseteq \pi\mathbb{Z}$) and so on.

Lemma

$\mathbb{A} \not\leq_T P$ for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ and $\mathbb{Z} \subseteq P \subseteq U$.

Proof: Let us assume that $\mathbb{A}$ is decidable by a machine in $\mathsf{M}_{\mathcal{A}}(P)$.

- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by an $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}(P)$.
- $\mathcal{M}$ can be modified:
  - The integers are enumerated and compared with the input.
  - If the input is a positive integer,
    then $\mathcal{M}$ can be simulated by a machine in $\mathsf{M}_{\mathcal{A}}$
    since
    - all queries of $\mathcal{M}$ are answered in the positive,
    - each order test can be simulated by means of equality tests.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by a machine in $\mathsf{M}_{\mathcal{A}}$.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N} = W_j$ for some j.
- $\Rightarrow$ By definition of $\mathbb{A}$ the assumption is wrong.

Lemma

$\mathbb{A} \not\leq_T P$ for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ and $\mathbb{Z} \subseteq P \subseteq U$.

Proof: Let us assume that $\mathbb{A}$ is decidable by a machine in $\mathsf{M}_{\mathcal{A}}(P)$.

- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by an $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}(P)$.
- $\mathcal{M}$ can be modified:
  - The integers are enumerated and compared with the input.
  - If the input is a positive integer, then $\mathcal{M}$ can be simulated by a machine in $\mathsf{M}_{\mathcal{A}}$ since
    - all queries of $\mathcal{M}$ are answered in the positive,
    - each order test can be simulated by means of equality tests.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by a machine in $\mathsf{M}_{\mathcal{A}}$.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N} = W_j$ for some j.
- $\Rightarrow$ By definition of $\mathbb{A}$ the assumption is wrong.

Lemma

$\mathbb{A} \not\leq_T P$ for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ and $\mathbb{Z} \subseteq P \subseteq U$.

Proof: Let us assume that $\mathbb{A}$ is decidable by a machine in $\mathsf{M}_{\mathcal{A}}(P)$.

- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by an $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}(P)$.
- $\mathcal{M}$ can be modified:
  - The integers are enumerated and compared with the input.
  - If the input is a positive integer,
    then $\mathcal{M}$ can be simulated by a machine in $\mathsf{M}_{\mathcal{A}}$
    since
    - all queries of $\mathcal{M}$ are answered in the positive,
    - each order test can be simulated by means of equality tests.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by a machine in $\mathsf{M}_{\mathcal{A}}$.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N} = W_j$ for some j.
- $\Rightarrow$ By definition of $\mathbb{A}$ the assumption is wrong.

A Generalization of the Friedberg-Muchnik Theorem

Lemma

$\mathbb{A} \not\leq_T P$ for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ and $\mathbb{Z} \subseteq P \subseteq U$.

Proof: Let us assume that $\mathbb{A}$ is decidable by a machine in $\mathrm{M}_{\mathcal{A}}(P)$.

- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by an $\mathcal{M} \in \mathrm{M}_{\mathcal{A}}(P)$.
- $\mathcal{M}$ can be modified:
  - The integers are enumerated and compared with the input.
  - If the input is a positive integer, then $\mathcal{M}$ can be simulated by a machine in $\mathrm{M}_{\mathcal{A}}$ since
    - all queries of $\mathcal{M}$ are answered in the positive,
    - each order test can be simulated by means of equality tests.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by a machine in $\mathrm{M}_{\mathcal{A}}$.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N} = W_j$ for some j.
- $\Rightarrow$ By definition of $\mathbb{A}$ the assumption is wrong.

A Generalization of the Friedberg-Muchnik Theorem

Lemma

$\mathbb{A} \not\leq_T P$ for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ and $\mathbb{Z} \subseteq P \subseteq U$.

Proof: Let us assume that $\mathbb{A}$ is decidable by a machine in $\mathsf{M}_{\mathcal{A}}(P)$.

- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by an $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}(P)$.
- $\mathcal{M}$ can be modified:
  - The integers are enumerated and compared with the input.
  - If the input is a positive integer,
    then $\mathcal{M}$ can be simulated by a machine in $\mathsf{M}_{\mathcal{A}}$
    since
    - all queries of $\mathcal{M}$ are answered in the positive,
    - each order test can be simulated by means of equality tests.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by a machine in $\mathsf{M}_{\mathcal{A}}$.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N} = W_j$ for some j.
- $\Rightarrow$ By definition of $\mathbb{A}$ the assumption is wrong.

A Generalization of the Friedberg-Muchnik Theorem

Lemma

$\mathbb{A} \not\leq_T P$ for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ and $\mathbb{Z} \subseteq P \subseteq U$.

Proof: Let us assume that $\mathbb{A}$ is decidable by a machine in $\mathsf{M}_{\mathcal{A}}(P)$.

- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by an $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}(P)$.
- $\mathcal{M}$ can be modified:
  - The integers are enumerated and compared with the input.
  - If the input is a positive integer,
    then $\mathcal{M}$ can be simulated by a machine in $\mathsf{M}_{\mathcal{A}}$
    since
    - all queries of $\mathcal{M}$ are answered in the positive,
    - each order test can be simulated by means of equality tests.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by a machine in $\mathsf{M}_{\mathcal{A}}$.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N} = W_j$ for some j.
- $\Rightarrow$ By definition of $\mathbb{A}$ the assumption is wrong.

A Generalization of the Friedberg-Muchnik Theorem

Lemma

$\mathbb{A} \not\leq_T P$ for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ and $\mathbb{Z} \subseteq P \subseteq U$.

Proof: Let us assume that $\mathbb{A}$ is decidable by a machine in $\mathsf{M}_{\mathcal{A}}(P)$.

- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by an $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}(P)$.
- $\mathcal{M}$ can be modified:
  - The integers are enumerated and compared with the input.
  - If the input is a positive integer,
    then $\mathcal{M}$ can be simulated by a machine in $\mathsf{M}_{\mathcal{A}}$
    since
    - all queries of $\mathcal{M}$ are answered in the positive,
    - each order test can be simulated by means of equality tests.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by a machine in $\mathsf{M}_{\mathcal{A}}$.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N} = W_j$ for some j.
- $\Rightarrow$ By definition of $\mathbb{A}$ the assumption is wrong.

A Generalization of the Friedberg-Muchnik Theorem

Lemma

$\mathbb{A} \not\leq_T P$ for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ and $\mathbb{Z} \subseteq P \subseteq U$.

Proof: Let us assume that $\mathbb{A}$ is decidable by a machine in $\mathrm{M}_{\mathcal{A}}(P)$.

- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by an $\mathcal{M} \in \mathrm{M}_{\mathcal{A}}(P)$.
- $\mathcal{M}$ can be modified:
  - The integers are enumerated and compared with the input.
  - If the input is a positive integer,
    then $\mathcal{M}$ can be simulated by a machine in $\mathrm{M}_{\mathcal{A}}$
    since
    - all queries of $\mathcal{M}$ are answered in the positive,
    - each order test can be simulated by means of equality tests.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by a machine in $\mathrm{M}_{\mathcal{A}}$.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N} = W_j$ for some j.
- $\Rightarrow$ By definition of $\mathbb{A}$ the assumption is wrong.

A Generalization of the Friedberg-Muchnik Theorem

Lemma

$\mathbb{A} \not\leq_T P$ for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ and $\mathbb{Z} \subseteq P \subseteq U$.

Proof: Let us assume that $\mathbb{A}$ is decidable by a machine in $\mathsf{M}_{\mathcal{A}}(P)$.

- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by an $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}(P)$.
- $\mathcal{M}$ can be modified:
  - The integers are enumerated and compared with the input.
  - If the input is a positive integer,
    then $\mathcal{M}$ can be simulated by a machine in $\mathsf{M}_{\mathcal{A}}$
    since
    - all queries of $\mathcal{M}$ are answered in the positive,
    - each order test can be simulated by means of equality tests.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by a machine in $\mathsf{M}_{\mathcal{A}}$.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N} = W_j$ for some j.
- $\Rightarrow$ By definition of $\mathbb{A}$ the assumption is wrong.

A Generalization of the Friedberg-Muchnik Theorem

Lemma

$\mathbb{A} \not\leq_T P$ for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ and $\mathbb{Z} \subseteq P \subseteq U$.

Proof: Let us assume that $\mathbb{A}$ is decidable by a machine in $\mathsf{M}_{\mathcal{A}}(P)$.

- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by an $\mathcal{M} \in \mathsf{M}_{\mathcal{A}}(P)$.
- $\mathcal{M}$ can be modified:
  - The integers are enumerated and compared with the input.
  - If the input is a positive integer,
    then $\mathcal{M}$ can be simulated by a machine in $\mathsf{M}_{\mathcal{A}}$
    since
    - all queries of $\mathcal{M}$ are answered in the positive,
    - each order test can be simulated by means of equality tests.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N}$ is semi-decidable by a machine in $\mathsf{M}_{\mathcal{A}}$.
- $\Rightarrow (\mathbb{R} \setminus \mathbb{A}) \cap \mathbb{N} = W_j$ for some j.
- $\Rightarrow$ By definition of $\mathbb{A}$ the assumption is wrong.

A Generalization of the Friedberg-Muchnik Theorem

Theorem

Let $\mathcal{A}$ be a structure containing only a finite number of constants and relations, the operations $f_1, \ldots, f_m$ of arities $\mu_1, \ldots, \mu_m$, and an effectively enumerable set $N \subseteq U$.

Let $F_0 = E_0 = N$, $F_i = \bigcup_{j \leq i} E_j$ where

$$E_{i+1} = \bigcup_{k=1}^{m} \{f_k(n_1, \ldots, n_{\mu_k}) \mid (n_1, \ldots, n_{\mu_k}) \in F_i^{\mu_k}\},$$

and let N be decidable on $E =_{\mathrm{df}} \bigcup_{i \geq 0} E_i$.

Moreover, let (a) or (b) be given.

(a) $P = \bigcup_{i \leq n} P_{i,1} \times \cdots \times P_{i,j_i}$ with $E \subseteq P_{i,k} \subseteq U$ for all $i \leq n, k \leq j_i$.

(b) $P \cap E^\infty$ is decidable for all inputs in E^∞.

Then, there is a semi-decidable $\mathbb{A} \subset N$ with $\mathbb{A} \not\preceq_T P$ and thus $\mathbb{H}_{\mathcal{A}} \not\preceq_T P$.

Summary

The examples show that extensive knowledge of classical recursion theory is a fundamental condition for a closer examination of algebraic computation models.

Thank you very much for your attention!