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 $\Delta^0_2$
  - the Limit Lemma

- The transfer of a further theorem from the Recursion Theory
  - a generalization of the Friedberg-Muchnik Theorem
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 $\Delta^0_2$
  - the Limit Lemma

The transfer of a further theorem from the Recursion Theory
  - a generalization of the Friedberg-Muchnik Theorem
Computation over Algebraic Structures

The Allowed Instructions

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:**
  
  $l$: $Z_j := f_k(Z_{j_1}, \ldots, Z_{j_m})$ (e.g. $l$: $Z_j := Z_{j_1} + Z_{j_2}$),
  
  $l$: $Z_j := d_k$,

- **Branching instructions:**
  
  $l$: if $Z_i = Z_j$ then goto $l_1$ else goto $l_2$,
  
  $l$: if $R_k(Z_{j_1}, \ldots, Z_{j_{nk}})$ then goto $l_1$ else goto $l_2$,

- **Copy instructions:**
  
  $l$: $Z_{I_j} := Z_{I_k}$,

- **Index instructions:**
  
  $l$: $I_j := 1$,
  
  $l$: $I_j := I_j + 1$,
  
  $l$: if $I_j = I_k$ then goto $l_1$ else goto $l_2$. 
Computation over Algebraic Structures
The Allowed Instructions

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:**
  
  $l$: $Z_j := f_k(Z_{j_1}, \ldots, Z_{j_m})$  \hspace{0.5cm} (e.g. $l$: $Z_j := Z_{j_1} + Z_{j_2}$),
  
  $l$: $Z_j := d_k$,

- **Branching instructions:**
  
  $l$: if $Z_i = Z_j$ then goto $l_1$ else goto $l_2$,
  
  $l$: if $R_k(Z_{j_1}, \ldots, Z_{j_{nk}})$ then goto $l_1$ else goto $l_2$,

- **Copy instructions:**
  
  $l$: $Z_{I_j} := Z_{I_k}$,

- **Index instructions:**
  
  $l$: $I_j := 1$,
  
  $l$: $I_j := I_j + 1$,
  
  $l$: if $I_j = I_k$ then goto $l_1$ else goto $l_2$. 
Computation over Algebraic Structures
The Allowed Instructions

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

- **Branching instructions:**
  
  $l$: if $Z_i = Z_j$ then goto $l_1$ else goto $l_2$,
  
  $l$: if $R_k(Z_{j_1}, \ldots, Z_{j_{n_k}})$ then goto $l_1$ else goto $l_2$,

- **Copy instructions:**
  
  $l$: $Z_{I_j} := Z_{I_k}$,

- **Index instructions:**
  
  $l$: $I_j := 1$,
  
  $l$: $I_j := I_j + 1$,
  
  $l$: if $I_j = I_k$ then goto $l_1$ else goto $l_2$. 
Computation over Algebraic Structures
The Allowed Instructions

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:**
  
  \[
  \begin{align*}
  l: & \quad Z_j := f_k(Z_{j1}, \ldots, Z_{jm_k}) \quad \text{(e.g. } l: \ Z_j := Z_{j1} + Z_{j2}), \\
  l: & \quad Z_j := d_k,
  \end{align*}
  \]

- **Branching instructions:**
  
  \[
  \begin{align*}
  l: & \quad \text{if } Z_i = Z_j \text{ then goto } l_1 \text{ else goto } l_2, \\
  l: & \quad \text{if } R_k(Z_{j1}, \ldots, Z_{jn_k}) \text{ then goto } l_1 \text{ else goto } l_2,
  \end{align*}
  \]

- **Copy instructions:**
  
  \[
  l: \ Z_{Ij} := Z_{Ik},
  \]

- **Index instructions:**
  
  \[
  \begin{align*}
  l: & \quad I_j := 1, \\
  l: & \quad I_j := I_j + 1, \\
  l: & \quad \text{if } I_j = I_k \text{ then goto } l_1 \text{ else goto } l_2.
  \end{align*}
  \]
Computation over Algebraic Structures
The Machines

- Input and output space: $U^\infty = \text{df } \bigcup_{i \geq 1} U^i$

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

- $M_A$ – machines over $A$

- $M_A(O)$ – machines using $O \subseteq U^\infty$ as oracle

Oracle instructions:

$l$: if $(Z_1, \ldots, Z_{I_1}) \in O$ then goto $l_1$ else goto $l_2$. 
Computation over Algebraic Structures
The Machines

- **Input and output space:** $U^\infty = \bigcup_{i \geq 1} U^i$

- **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** $Z_1, \ldots, Z_{I_1}$.

- $M_A$ – machines over $A$

- $M_A(\mathcal{O})$ – 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 $l_2.$
Computation over Algebraic Structures
The Machines

- Input and output space: \( U^\infty = \bigcup_{i \geq 1} U^i \)

- 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 \( Z_1, \ldots, Z_{I_1} \).

- \( M_\mathcal{A} \) – machines over \( \mathcal{A} \)

- \( M_\mathcal{A}(\mathcal{O}) \) – 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 \( l_2 \).
Computation over Algebraic Structures
The Machines

- **Input and output space:** \( U^\infty = \bigcup_{i \geq 1} U^i \)

- **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 \( Z_1, \ldots, Z_{I_1}. \)

- \( M_\mathcal{A} \) – machines over \( \mathcal{A} \)

- \( M_\mathcal{A}(O) \) – machines using \( O \subseteq U^\infty \) as oracle

**Oracle instructions:**

\[
l: \text{ if } (Z_1, \ldots, Z_{I_1}) \in O \text{ then goto } l_1 \text{ else goto } l_2.
\]
Computation over Algebraic Structures
The Machines

- Input and output space: \( U^\infty = \bigcup_{i \geq 1} U^i \)

- 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 \( Z_1, \ldots, Z_{I_1}. \)

- \( M_A \) – machines over \( A \)

- \( M_A(O) \) – machines using \( O \subseteq U^\infty \) as oracle

**Oracle instructions:**

\[
l: \text{if } (Z_1, \ldots, Z_{I_1}) \in O \text{ then goto } l_1 \text{ else goto } l_2.
\]
Computation over Algebraic Structures
The Halting Problem

\[ H_A = \{(\vec{x} \cdot c_M) \mid \vec{x} \in U^\infty \land M \in M_A \land M(\vec{x}) \downarrow\} \]

where

\[ \vec{x} = (x_1, \ldots, x_n) \]
\[ c_M = \text{code}(M) = (s_1, \ldots, s_m) \]
\[ (\vec{x} \cdot c_M) = (x_1, \ldots, x_n, s_1, \ldots, s_m) \]

\[ M(\vec{x}) \downarrow \overset{\sim}{=} M \text{ halts on } \vec{x} \]

\[ H_A \in \text{REC}_A \text{ if } A \text{ is a structure of finite signature} \]
\[ H_A \not\in \text{DEC}_A \]

\[ \text{REC}_A \text{ – recognizable (semi-decidable) problems} \]
\[ \text{DEC}_A \text{ – decidable problems} \]

\[ H_O^A = \{(\vec{x} \cdot c_M) \mid \vec{x} \in U^\infty \land M \in M_A(O) \land M(\vec{x}) \downarrow\} \]

\[ H_O^A \not\in \text{DEC}_A^O \]
Computation over Algebraic Structures
The Halting Problem

$H_A = \{ (\vec{x} \cdot c_M) \mid \vec{x} \in U^\infty \land M \in M_A \land M(\vec{x}) \downarrow \}$

where

$\vec{x} = (x_1, \ldots, x_n)$

$c_M = \text{code}(M) = (s_1, \ldots, s_m)$

$(\vec{x} \cdot c_M) = (x_1, \ldots, x_n, s_1, \ldots, s_m)$

$M(\vec{x}) \downarrow \overset{\text{def}}{=} M \text{ halts on } \vec{x}$

$H_A \in \text{REC}_A$ if $A$ is a structure of finite signature
$H_A \not\in \text{DEC}_A$

$\text{REC}_A$ – recognizable (semi-decidable) problems
$\text{DEC}_A$ – decidable problems

$H^O_A = \{ (\vec{x} \cdot c_M) \mid \vec{x} \in U^\infty \land M \in M_A(O) \land M(\vec{x}) \downarrow \}$

$H^O_A \not\in \text{DEC}^O_A$
Computation over Algebraic Structures

The Halting Problem

\[ H_A = \{(\vec{x} \cdot c_M) \mid \vec{x} \in U^\infty \land M \in M_A \land M(\vec{x}) \downarrow\} \]

where

\[ \vec{x} = (x_1, \ldots, x_n) \]
\[ c_M = \text{code}(M) = (s_1, \ldots, s_m) \]
\[ (\vec{x} \cdot c_M) = (x_1, \ldots, x_n, s_1, \ldots, s_m) \]

\[ M(\vec{x}) \downarrow \equiv M \text{ halts on } \vec{x} \]

\[ H_A \in \text{REC}_A \text{ if } A \text{ is a structure of finite signature} \]
\[ H_A \not\in \text{DEC}_A \]

**REC** \(\_A\) – recognizable (semi-decidable) problems

**DEC** \(\_A\) – decidable problems

\[ H^O_A = \{(\vec{x} \cdot c_M) \mid \vec{x} \in U^\infty \land M \in M_A(O) \land M(\vec{x}) \downarrow\} \]

\[ H^O_A \not\in \text{DEC}^O_A \]
Computation over Algebraic Structures

The Halting Problem

\[ H_A = \{ (\vec{x} \cdot c_M) \mid \vec{x} \in U^\infty \land M \in M_A \land M(\vec{x}) \downarrow \} \]

where

\[ \vec{x} = (x_1, \ldots, x_n) \]
\[ c_M = \text{code}(M) = (s_1, \ldots, s_m) \]
\[ (\vec{x} \cdot c_M) = (x_1, \ldots, x_n, s_1, \ldots, s_m) \]

\[ M(\vec{x}) \downarrow \overset{\hat{\cdot}}{=} M \text{ halts on } \vec{x} \]

\[ H_A \in \text{REC}_A \text{ if } A \text{ is a structure of finite signature} \]
\[ H_A \notin \text{DEC}_A \]

\[ \text{REC}_A \text{ – recognizable (semi-decidable) problems} \]
\[ \text{DEC}_A \text{ – decidable problems} \]

\[ H^O_A = \{ (\vec{x} \cdot c_M) \mid \vec{x} \in U^\infty \land M \in M_A(O) \land M(\vec{x}) \downarrow \} \]

\[ H^O_A \notin \text{DEC}^O_A \]
The Turing Reduction over Structures \( \mathcal{A} \)

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

\[ P \leq_T Q \quad \text{\( P \) is easier than \( Q \),} \]
\[ \quad \text{\( P \) is decidable by a machine in \( M_{\mathcal{A}}(Q) \).} \]

\[ P \not\leq_T Q \quad \text{\( P \) is strictly easier than \( Q \),} \]
\[ \quad \text{\( Q \) cannot be decided by a machine in \( M_{\mathcal{A}}(P) \).} \]

- \( \Rightarrow \) For the Halting Problem:

\[ P \in \text{REC}_{\mathcal{A}} \Rightarrow P \preceq_1 \mathbb{H}_{\mathcal{A}} \quad \text{(one-one reduction over \( \mathcal{A} \))} \]
\[ \Rightarrow P \preceq_T \mathbb{H}_{\mathcal{A}} \]
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 $M_\mathcal{A}(Q)$.

  - $P \npreceq_T Q$  
    
    $P$ is strictly easier than $Q$, 
    $Q$ cannot be decided by a machine in $M_\mathcal{A}(P)$.

$\Rightarrow$ For the Halting Problem:

- $P \in \text{REC}_\mathcal{A} \Rightarrow P \preceq_1 H_\mathcal{A}$ (one-one reduction over $\mathcal{A}$)
  
  $\Rightarrow P \preceq_T H_\mathcal{A}$
The Turing Reduction
over Structures $\mathcal{A}$

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

  $P \preceq_T Q \quad P$ is easier than $Q$,
  $P$ is decidable by a machine in $M_{\mathcal{A}}(Q)$.

  $P \not\preceq_T Q \quad P$ is strictly easier than $Q$,
  $Q$ cannot be decided by a machine in $M_{\mathcal{A}}(P)$.

⇒ For the Halting Problem:

  $P \in \text{REC}_{\mathcal{A}} \Rightarrow P \preceq_1 H_{\mathcal{A}}$ (one-one reduction over $\mathcal{A}$)
  $\Rightarrow P \preceq_T H_{\mathcal{A}}$
A Hierarchy
(Analogously to the Arithmetical Hierarchy)

- **Definition (semantically by deterministic machines):**
  \( A \) is fixed.

\[
\begin{align*}
\Sigma_0^0 &= \text{DEC}_A, \\
\Pi_n^0 &= \{ U^\infty \setminus P \mid P \in \Sigma_n^0 \}, \\
\Delta_n^0 &= \Sigma_n^0 \cap \Pi_n^0, \\
\Sigma_{n+1}^0 &= \{ P \subseteq U^\infty \mid (\exists Q \in \Sigma_n^0)(P \subseteq_1 H^n_A) \}.
\end{align*}
\]

- **The first level:**

\[
\begin{align*}
\Sigma_1^0 &= \text{REC}_A = \{ P \subseteq U^\infty \mid P \subseteq_1 H_A \}, \\
\Pi_1^0 &= \{ P \subseteq U^\infty \mid P \subseteq_1 U^\infty \setminus H_A \}, \\
\Delta_1^0 &= \text{DEC}_A = \{ P \subseteq U^\infty \mid P \subseteq_T \emptyset \},
\end{align*}
\]
Definition (semantically by deterministic machines): 

\( A \) is fixed.

\[
\begin{align*}
\Sigma_0^0 &= \text{DEC}_A, \\
\Pi_n^0 &= \{ U^\infty \setminus P \mid P \in \Sigma_n^0 \}, \\
\Delta_n^0 &= \Sigma_n^0 \cap \Pi_n^0, \\
\Sigma_{n+1}^0 &= \{ P \subseteq U^\infty \mid (\exists Q \in \Sigma_n^0)(P \preceq_1 H^O_A) \}.
\end{align*}
\]

The first level:

\[
\begin{align*}
\Sigma_1^0 &= \text{REC}_A = \{ P \subseteq U^\infty \mid P \preceq_1 H_A \}, \\
\Pi_1^0 &= \{ P \subseteq U^\infty \mid P \preceq_1 U^\infty \setminus H_A \}, \\
\Delta_1^0 &= \text{DEC}_A = \{ P \subseteq U^\infty \mid P \preceq_T \emptyset \}.
\end{align*}
\]
Definition (semantically by deterministic machines): 
\( \mathcal{A} \) is fixed.

\[
\begin{align*}
\Sigma_0^0 &= \text{DEC}_\mathcal{A}, \\
\Pi_n^0 &= \{ U^\infty \setminus P \mid P \in \Sigma_n^0 \}, \\
\Delta_n^0 &= \Sigma_n^0 \cap \Pi_n^0, \\
\Sigma_{n+1}^0 &= \{ P \subseteq U^\infty \mid (\exists Q \in \Sigma_n^0)(P \preceq_1 \mathcal{H}^Q_\mathcal{A}) \}.
\end{align*}
\]

The first level:

\[
\begin{align*}
\Sigma_1^0 &= \text{REC}_\mathcal{A} = \{ P \subseteq U^\infty \mid P \preceq_1 \mathcal{H}_\mathcal{A} \}, \\
\Pi_1^0 &= \{ P \subseteq U^\infty \mid P \preceq_1 U^\infty \setminus \mathcal{H}_\mathcal{A} \}, \\
\Delta_1^0 &= \text{DEC}_\mathcal{A} = \{ P \subseteq U^\infty \mid P \preceq_T \emptyset \},
\end{align*}
\]
A Hierarchy
(Analogously to the Arithmetical Hierarchy)

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

\[
\begin{align*}
\Sigma^0_0 &= \text{DEC}_\mathcal{A}, \\
\Pi^0_n &= \{ U^\infty \setminus P \mid P \in \Sigma^0_n \}, \\
\Delta^0_n &= \Sigma^0_n \cap \Pi^0_n, \\
\Sigma^0_{n+1} &= \{ P \subseteq U^\infty \mid (\exists Q \in \Sigma^0_n)(P \preceq_1 \mathcal{H}_\mathcal{A}^Q) \}.
\end{align*}
\]

The second level:

\[
\begin{align*}
\Sigma^0_2 &= \text{REC}^{\mathcal{H}_\mathcal{A}}_\mathcal{A} = \{ P \subseteq U^\infty \mid P \preceq_1 \mathcal{H}_\mathcal{A} \}, \\
\Pi^0_2 &= \{ P \subseteq U^\infty \mid P \preceq_1 U^\infty \setminus \mathcal{H}_\mathcal{A} \}, \\
\Delta^0_2 &= \text{DEC}^{\mathcal{H}_\mathcal{A}}_\mathcal{A} = \{ P \subseteq U^\infty \mid P \preceq_T \mathcal{H}_\mathcal{A} \}.
\end{align*}
\]
Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.

$\chi_P$ – 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 $\{(n . \vec{x}) | n \in \mathbb{N} \& \vec{x} \in U^\infty\}$ 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 $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \leq_T \mathbb{H}_A \}$

- Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.
- $\chi_P$ – 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 \cdot \vec{x}) \mid n \in \mathbb{N} \& \vec{x} \in U^\infty\} such that $\chi_P(\vec{x}) = \lim_{s \to \infty} g(s \cdot \vec{x})$.

**Lemma (First Part of Limit Lemma)**

If (1), then (2).

**Lemma (Second Part of Limit Lemma)**

If (2), then (1).
Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.

$\chi_P$ – 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 $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \preceq_T H_A \}$

- Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.
- $\chi_P$ – 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 $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \leq_T H_A \}$

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

**Lemma (First Part of Limit Lemma)**

If (1), then (2).

**Lemma (Second Part of Limit Lemma)**

If (2), then (1).
A Characterization of $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \leq_T H_A \}$

- Let $\mathcal{A}$ contain an effectively enumerable set denoted by $\mathbb{N}$.
- $\chi_P$ – 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 $\{(n, \vec{x}) \mid n \in \mathbb{N} \land \vec{x} \in U^\infty\}$ 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 $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \leq_T \mathbb{H}_A \}$

Lemma (First Part of Limit Lemma)

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

Proof. Let $M \in M_A(\mathbb{H}_A)$ 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 $(\vec{y}^{(i)} \cdot c_{L_i}) \in \mathbb{H}_A$ executed by $M$ on input $\vec{x}$.
- $\Rightarrow$
- $\beta_i = 0$ iff $L_i(\vec{y}^{(i)}) \uparrow$,
- $\beta_i = t > 0$ iff $L_i(\vec{y}^{(i)}) \uparrow^{t-1}$ and $L_i(\vec{y}^{(i)}) \downarrow^t$. 
Lemma (First Part of Limit Lemma)

If $P \subseteq U^\infty$ is in $\Delta_2^0$, then there is a computable function $g : U^\infty \to \{0, 1\}$ defined on $\{(n \cdot \vec{x}) \mid n \in \mathbb{N} \land \vec{x} \in U^\infty\}$ such that $
abla_P(\vec{x}) = \lim_{s \to \infty} g(s \cdot \vec{x})$.

Proof. Let $M \in M_A(\mathbb{H}_A)$ 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 $(\vec{y}^{(i)} \cdot c_{L_i}) \in \mathbb{H}_A$? executed by $M$ on input $\vec{x}$.

- $\Rightarrow$

- $\beta_i = 0$ iff $L_i(\vec{y}^{(i)}) \uparrow$,
- $\beta_i = t > 0$ iff $L_i(\vec{y}^{(i)}) \uparrow^{t-1}$ and $L_i(\vec{y}^{(i)}) \downarrow^t$. 
A Characterization of $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \preceq_T H_A \}$

Lemma (First Part of Limit Lemma)

If $P \subseteq U^\infty$ is in $\Delta^0_2$, 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 $M \in M_A(H_A)$ 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 $(\vec{y}^{(i)} . c_{L_i}) \in H_A$? executed by $M$ on input $\vec{x}$.

- $\Rightarrow$

  - $\beta_i = 0$ iff $L_i(\vec{y}^{(i)}) \uparrow$,
  - $\beta_i = t > 0$ iff $L_i(\vec{y}^{(i)}) \uparrow^{t-1}$ and $L_i(\vec{y}^{(i)}) \downarrow^t$. 

Lemma (First Part of Limit Lemma)

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

Proof. Let $M \in M_A(\mathbb{H}_A)$ 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
  $(\vec{y}^{(i)} \cdot c_{L_i}) \in \mathbb{H}_A$? executed by $M$ on input $\vec{x}$.

- $\Rightarrow$

- $\beta_i = 0$ iff $L_i(\vec{y}^{(i)}) \uparrow$,

- $\beta_i = t > 0$ iff $L_i(\vec{y}^{(i)}) \uparrow^{t-1}$ and $L_i(\vec{y}^{(i)}) \downarrow^t$. 

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

Lemma (First Part of Limit Lemma)

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

Proof. Let $M \in M_A(\mathbb{H}_A)$ 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 $(\vec{y}^{(i)} \cdot c_{L_i}) \in \mathbb{H}_A$? executed by $M$ on input $\vec{x}$.

$\Rightarrow$

- $\beta_i = 0$ iff $L_i(\vec{y}^{(i)}) \uparrow$,
- $\beta_i = t > 0$ iff $L_i(\vec{y}^{(i)}) \uparrow^{t-1}$ and $L_i(\vec{y}^{(i)}) \downarrow^t$. 
A Characterization of $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \preceq_T H_A \}$

- Let $g$ be computed by $N \in M_A$ executing:
  - Input $(s, \bar{x}) \in U^\infty$;
  - if $s \in \mathbb{N}$, then simulate $s$ instructions of $M$, but, instead of
    
    \[ \text{if } (\bar{y}^{(s,i)} \cdot c_{L_s,i}) \in H_A \text{ then goto to } l_1 \text{ else goto } l_2, \]
    
    use
    
    \[ \text{if } L_{s,i}(\bar{y}^{(s,i)}) \downarrow^s \text{ then goto to } l_1 \text{ else goto } l_2. \]
  - If the output of $M$ is not reached within $s$ steps of $M$, then output 0.

- $s \in \mathbb{N}, \bar{x} \in U^\infty \Rightarrow$ let $\beta_1^{(s)}, \ldots, \beta_{m_s}^{(s)} \leq s \ (m_s \leq s)$ with
  \[ \beta_i^{(s)} \neq 0 \Rightarrow L_{s,i}(\bar{y}^{(s,i)}) \uparrow^{\beta_i^{(s)}}_{i-1} \text{ and } L_{s,i}(\bar{y}^{(s,i)}) \downarrow^{\beta_i^{(s)}}. \]

- $s \in \mathbb{N}, \bar{x} \in U^\infty \Rightarrow$ there are $0 = s_0 < s_1 \leq s_2 \leq \cdots \leq s_k$ such that
  \[ (\beta_1, \beta_2, \ldots, \beta_i) = (\beta_1^{s_i}, \beta_2^{s_i}, \ldots, \beta_i^{s_i}) \quad \text{for } i \leq m_s, \]
  \[ (\beta_1, \beta_2, \ldots, \beta_k) = (\beta_1^{s}, \beta_2^{s}, \ldots, \beta_k^{s}) \quad \text{for } s \geq s_k. \]

- $\bar{x} \in U^\infty \Rightarrow$ there is an $s_{\bar{x}}$ such that $N$ outputs the same value as $M$ on $(s, \bar{x})$ for all $s \geq s_{\bar{x}}$.  
A Characterization of $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \leq_T H_A \}$

- Let $g$ be computed by $N \in M_A$ executing:
  - Input $(s \cdot \vec{x}) \in U^\infty$;
  - if $s \in \mathbb{N}$, then simulate $s$ instructions of $M$, but, instead of
    
    $$
    \text{if } (\vec{y}^{(s,i)} \cdot c_{L,s,i}) \in H_A \text{ then goto to } l_1 \text{ else goto } l_2,
    $$

    use
    
    $$
    \text{if } L_{s,i}(\vec{y}^{(s,i)}) \downarrow^s \text{ then goto to } l_1 \text{ else goto } l_2.
    $$

- If the output of $M$ is not reached within $s$ steps of $M$, 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
  $$
  \beta_i^{(s)} \neq 0 \Rightarrow L_{s,i}(\vec{y}^{(s,i)}) \uparrow^{\beta_i^{(s)}-1} \text{ and } L_{s,i}(\vec{y}^{(s,i)}) \downarrow^{\beta_i^{(s)}}.
  $$

- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ there are $0 = s_0 < s_1 \leq s_2 \leq \cdots \leq s_k$ such that
  $$
  (\beta_1, \beta_2, \ldots, \beta_i) = (\beta_1^{s_i}, \beta_2^{s_i}, \ldots, \beta_i^{s_i}) \quad \text{for } i \leq m_{s_i},
  $$
  
  $$
  (\beta_1, \beta_2, \ldots, \beta_k) = (\beta_1^{s}, \beta_2^{s}, \ldots, \beta_k^{s}) \quad \text{for } s \geq s_k.
  $$

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

$\square$
A Characterization of $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \preceq_T H_A \}$

- Let $g$ be computed by $N \in M_A$ executing:
  - Input $(s . \vec{x}) \in U^\infty$;
  - if $s \in \mathbb{N}$, then simulate $s$ instructions of $M$, but, instead of
    
    $$\text{if } (\vec{y}^{(s,i)} . c_{L,s,i}) \in H_A \text{ then goto to } l_1 \text{ else goto } l_2,$$

    use
    $$\text{if } L_{s,i}(\vec{y}^{(s,i)}) \downarrow^s \text{ then goto to } l_1 \text{ else goto } l_2.$$

  - If the output of $M$ is not reached within $s$ steps of $M$, 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
  
  $$\beta_i^{(s)} \neq 0 \Rightarrow L_{s,i}(\vec{y}^{(s,i)}) \uparrow \beta_i^{(s)}-1 \text{ and } L_{s,i}(\vec{y}^{(s,i)}) \downarrow \beta_i^{(s)}.$$

- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ there are $0 = s_0 < s_1 \leq s_2 \leq \cdots \leq s_k$ such that
  
  $$\begin{align*}
  (\beta_1, \beta_2, \ldots, \beta_i) &= (\beta_1^{s_i}, \beta_2^{s_i}, \ldots, \beta_i^{s_i}) \quad \text{for } i \leq m_{s_i}, \\
  (\beta_1, \beta_2, \ldots, \beta_k) &= (\beta_1^{s}, \beta_2^{s}, \ldots, \beta_k^{s}) \quad \text{for } s \geq s_k.
  \end{align*}$$

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

- \[ \begin{align*}
  \beta_1^{(s)} &= \beta_1^{(s)} \quad \text{for } i \leq m_{s_i}, \\
  \beta_1^{(s)} &= \beta_1^{(s)} \quad \text{for } s \geq s_k.
  \end{align*} \]
A Characterization of $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \preceq_T H_A \}$

- Let $g$ be computed by $N \in M_A$ executing:
  - Input $(s \cdot \vec{x}) \in U^\infty$;
  - if $s \in \mathbb{N}$, then simulate $s$ instructions of $M$, but, instead of
    
    $$\text{if } (\vec{y}^{(s,i)} \cdot c_{L_{s,i}}) \in H_A \text{ then goto to } l_1 \text{ else goto } l_2,$$

    use
    
    $$\text{if } L_{s,i} (\vec{y}^{(s,i)}) \uparrow^s \text{ then goto to } l_1 \text{ else goto } l_2.$$

  - If the output of $M$ is not reached within $s$ steps of $M$, then output 0.

- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow \text{let } \beta_1^{(s)}, \ldots, \beta_{m_s}^{(s)} \leq s (m_s \leq s)$ with
  
  $$\beta_i^{(s)} \neq 0 \Rightarrow L_{s,i} (\vec{y}^{(s,i)}) \uparrow^{\beta_i^{(s)}-1} \text{ and } L_{s,i} (\vec{y}^{(s,i)}) \downarrow^{\beta_i^{(s)}}.$$

- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ there are $0 = s_0 < s_1 \leq s_2 \leq \cdots \leq s_k$ such that
  
  $$(\beta_1, \beta_2, \ldots, \beta_i) = (\beta_1^{s_i}, \beta_2^{s_i}, \ldots, \beta_i^{s_i}) \text{ for } i \leq m_{s_i},$$
  
  $$(\beta_1, \beta_2, \ldots, \beta_k) = (\beta_1^s, \beta_2^s, \ldots, \beta_k^s) \text{ for } s \geq s_k.$$ 

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


A Characterization of $\Delta_2^0 = \{ P \subseteq U^\infty \mid P \preceq_T H_A \}$

- Let $g$ be computed by $\mathcal{N} \in M_A$ executing:
  - Input $(s \cdot \vec{x}) \in U^\infty$;
  - if $s \in \mathbb{N}$, then simulate $s$ instructions of $\mathcal{M}$, but, instead of
    
    $$if \ (\vec{y}^{(s,i)} \cdot c_{L,s,i}) \in H_A \ then \ goto \ to \ l_1 \ else \ goto \ l_2,$$
    
    use
    
    $$if \ L_{s,i}(\vec{y}^{(s,i)}) \downarrow^s \ then \ goto \ to \ l_1 \ else \ goto \ l_2.$$
  - 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
  
  $$\beta_i^{(s)} \neq 0 \Rightarrow \ L_{s,i}(\vec{y}^{(s,i)}) \uparrow^{\beta_i^{(s)} - 1} \ and \ L_{s,i}(\vec{y}^{(s,i)}) \downarrow^{\beta_i^{(s)}}.$$

- $s \in \mathbb{N}, \vec{x} \in U^\infty \Rightarrow$ there are $0 = s_0 < s_1 \leq s_2 \leq \cdots \leq s_k$ such that
  
  $$(\beta_1, \beta_2, \ldots, \beta_i) = (\beta_1^{s_i}, \beta_2^{s_i}, \ldots, \beta_i^{s_i}) \quad \text{for } i \leq m_s,$$

  $$(\beta_1, \beta_2, \ldots, \beta_k) = (\beta_1^s, \beta_2^s, \ldots, \beta_k^s) \quad \text{for } s \geq s_k.$$

- $\vec{x} \in U^\infty \Rightarrow$ there is an $s_{\vec{x}}$

  such that $\mathcal{N}$ outputs the same value as $\mathcal{M}$ on $(s \cdot \vec{x})$ for all $s \geq s_{\vec{x}}$. 
A Characterization of $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \leq_T \mathbb{H}_A \}$

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 $\Delta^0_2$.

Proof. Let $g$ be computed by $N \in M_A$ and let $M \in M_A(\mathbb{H}_A)$ execute:

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

$M$ decides $P$. 
A Characterization of $\Delta^0_2 = \{ P \subseteq U^\infty \mid P \preceq_T \mathbb{H}_A \}$

**Lemma (Second Part of Limit Lemma)**

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

**Proof.** Let $g$ be computed by $\mathcal{N} \in M_A$ and let $\mathcal{M} \in M_A(\mathbb{H}_A)$ execute:

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

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

Lemma (Second Part of Limit Lemma)

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

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

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

$\mathcal{M}$ decides $P$. 

Lemma (Second Part of Limit Lemma)

If there is a computable function \( g : U^\infty \rightarrow \{0, 1\} \) defined on \( \{(n \cdot \vec{x}) \mid n \in \mathbb{N} \& \vec{x} \in U^\infty\} \) such that \( \chi_P(\vec{x}) = \lim_{s \to \infty} g(s \cdot \vec{x}) \), then \( P \subseteq U^\infty \) is in \( \Delta_2^0 \).

Proof. Let \( g \) be computed by \( \mathcal{N} \in M_A \) and let \( \mathcal{M} \in M_A(H_A) \) execute:

- Input \( \vec{x} \in U^\infty \);
- Let \( s = 1 \);
- 1:
  - Ask \( ((s \cdot \vec{x}) \cdot c_L) \in H_A? \) where
    \( L : \text{Input } (s \cdot \vec{x}) \);
    Halt if there is a \( k \geq s \) such that \( g(s \cdot \vec{x}) \neq g(k \cdot \vec{x}) \).
    If \( L(s \cdot \vec{x}) \downarrow \),
    then \( s := s + 1 \) and goto 1
    else compute \( g(s \cdot \vec{x}) \) by simulating \( \mathcal{N} \) and output \( g(s \cdot \vec{x}) \).

\( \mathcal{M} \) decides \( P \).
Summary: $\Delta^0_2$ and the Limit Lemma

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

**Lemma (Limit Lemma)**

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

$\Rightarrow \exists H \in \text{REC}_\mathcal{A}$.

We construct an $\mathcal{A} \subset \mathbb{N}$ with
- $\mathcal{A} \not\preceq_T H$,
- $\mathcal{A} \not\preceq_T P$ for many $P \subseteq U^\infty$,
- $\mathcal{A} \not\preceq_T P$ for many $P \in \Delta^0_2$.

$\Rightarrow \exists H \preceq_T \not P$. 

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.

$\Rightarrow \mathbb{H}_\mathcal{A} \in \text{REC}_\mathcal{A}$.

We construct an $\mathcal{A} \subset \mathbb{N}$ with
- $\mathcal{A} \not\preceq_T \mathbb{H}_\mathcal{A}$,
- $\mathcal{A} \not\preceq_T P$ for many $P \subseteq U^\infty$,
- $\mathcal{A} \not\preceq_T P$ for many $P \in \Delta^0_2$.
- $\Rightarrow \mathbb{H}_\mathcal{A} \not\preceq_T P$. 
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.

$\implies H_{\mathcal{A}} \in \text{REC}_{\mathcal{A}}$.

We construct an $\mathcal{A} \subset \mathbb{N}$ with

- $\mathcal{A} \not\preceq_T H_{\mathcal{A}}$,
- $\mathcal{A} \not\preceq_T P$ for many $P \subseteq \mathcal{U}_\infty$,
- $\mathcal{A} \not\preceq_T P$ for many $P \in \Delta^0_2$.

$\implies H_{\mathcal{A}} \not\preceq_T P$. 
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:
- \( H_M \cap \mathbb{N} \) (for the halting set \( H_M \) of \( M \in M_A \)) is
effectively enumerable
- a halting set of a machine in \( M_A \).

For any \( \mathcal{O} \subseteq \mathbb{N} \),
we can list \( M_A(\mathcal{O}) : M_1^\mathcal{O}, M_2^\mathcal{O}, \ldots \).
(The index is the code of the corresponding program.)

We can list \( M_A : \mathcal{N}_1, \mathcal{N}_2, \ldots \).
\( \bar{N}_i \) enumerating all positive integers \( n_{i,1}, n_{i,2}, \ldots \in H_{\mathcal{N}_i} \).
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:
- \( H_M \cap \mathbb{N} \) (for the halting set \( H_M \) of \( M \in M_A \)) is
  - effectively enumerable
  - a halting set of a machine in \( M_A \).

- For any \( O \subseteq \mathbb{N} \),
  we can list \( M_A(O) \): \( M^O_1, M^O_2, \ldots \).
  (The index is the code of the corresponding program.)
- We can list \( M_A \): \( \mathcal{N}_1, \mathcal{N}_2, \ldots \).
- \( \mathcal{N}_i \) enumerating all positive integers \( n_{i,1}, n_{i,2}, \ldots \in H_{\mathcal{N}_i} \).
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:

- $H_M \cap \mathbb{N}$ (for the halting set $H_M$ of $M \in M_A$) is
  - effectively enumerable
  - a halting set of a machine in $M_A$.

- For any $\mathcal{O} \subseteq \mathbb{N}$, we can list $M_A(\mathcal{O})$: $M_1^\mathcal{O}, M_2^\mathcal{O}, \ldots$. (The index is the code of the corresponding program.)

- We can list $M_A$: $N_1, N_2, \ldots$

- $\bar{N}_i$ enumerating all positive integers $n_{i,1}, n_{i,2}, \ldots \in H_{N_i}$. 
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:
- \( H_M \cap \mathbb{N} \) (for the halting set \( H_M \) of \( M \in M_A \)) is
  - effectively enumerable
  - a halting set of a machine in \( M_A \).

For any \( \mathcal{O} \subseteq \mathbb{N} \),
we can list \( M_A(\mathcal{O}) : M_1^\mathcal{O}, M_2^\mathcal{O}, \ldots \)
(The index is the code of the corresponding program.)

We can list \( M_A : N_1, N_2, \ldots \)
- \( \bar{N}_i \) enumerating all positive integers \( n_{i,1}, n_{i,2}, \ldots \in H_{\bar{N}_i} \).
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:

- \( H_M \cap \mathbb{N} \) (for the halting set \( H_M \) of \( M \in M_A \)) is
  - effectively enumerable
  - a halting set of a machine in \( M_A \).

For any \( O \subseteq \mathbb{N} \),

we can list \( M_A(O) \): \( M_1^O, M_2^O, \ldots \)

(The index is the code of the corresponding program.)

We can list \( M_A \): \( \mathcal{N}_1, \mathcal{N}_2, \ldots \)

\( \tilde{\mathcal{N}}_i \) enumerating all positive integers \( n_{i,1}, n_{i,2}, \ldots \in H_{\mathcal{N}_i} \).
Let $\mathbb A = \bigcup_{s \geq 0} \mathbb A_s$ be defined in stages. $\mathbb A_0 = \emptyset$, $s \geq 0$.

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

where, for any $j \leq s$,

$$a(j, s) \begin{cases} \text{greatest integer used in a query by} \\ \mathcal M_j^{\mathbb A_s} \text{ on } j \text{ within } s \text{ steps} \\ 0 \end{cases} \quad \text{if } \mathcal M_j^{\mathbb A_s}(j) \downarrow^s,$$

$$\text{if } \mathcal M_j^{\mathbb A_s}(j) \uparrow^s.$$

$W_{i,s}$ is the set of integers computed by $\bar N_i$ on $s$ within $s$ steps.

If $I_s \neq \emptyset$, then let

$$i_s = \min I_s,$$

$$x_{i_s} = \min \{ x \in W_{i_s, s} \mid 2i_s < x \land (\forall j \leq i_s)(a(j, s) < x) \},$$

$$\mathbb A_{s+1} = \begin{cases} \mathbb A_s \quad \text{if } I_s = \emptyset \\ \mathbb A_s \cup \{ x_{i_s} \} \quad \text{otherwise.}\end{cases}$$
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 $j \leq s$,

1. $a(j, s) \begin{cases} \text{greatest integer used in a query by} \\ \mathcal{M}^{\mathbb{A}_s}_j \text{ on } j \text{ within } s \text{ steps} \\ 0 \end{cases}$ if $\mathcal{M}^{\mathbb{A}_s}_j(j) \downarrow^s$, 
   if $\mathcal{M}^{\mathbb{A}_s}_j(j) \uparrow^s$.

2. $W_{i,s}$ is the set of integers computed by $\tilde{N}_i$ on $s$ within $s$ steps.

If $I_s \neq \emptyset$, then let

$$i_s = \min I_s,$$
$$x_{i_s} = \min \{x \in W_{i,s} \mid 2i_s < x \& (\forall j \leq i_s)(a(j, s) < x)\},$$
$$\mathbb{A}_{s+1} = \begin{cases} \mathbb{A}_s & \text{if } I_s = \emptyset \\ \mathbb{A}_s \cup \{x_{i_s}\} & \text{otherwise}. \end{cases}$$
The properties of \( A \), for instance, for \( A = (\mathbb{R}; 0, 1; +, -, \cdots; \leq) \):

- \( A \) is effectively enumerable by an machine in \( M_A \).
  \[ \Rightarrow A \preceq_1 \mathbb{H}_A. \]

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

- Conditions for lowness for all \( n > 0 \):
  \( (N_n) \) If \( M_n^{A_t}(n) \downarrow_t \) for infinitely many \( t \), then \( M_n^A(n) \downarrow. \)
  \[ \Rightarrow \text{Conditions for simplicity for all } n > 0: \]
  \( (P_n) \) If \( W_n = \bigcup_{i \geq 1} W_{n,i} \) is infinite, then \( A \cap W_n \neq \emptyset. \)
  \[ \Rightarrow A^C \not\preceq_1 \mathbb{H}_A. \]

- \( K^A \preceq_T K^0 \) where \( K^O = \{ k_M \mid M \in M^1_A(O) \land M(k_M) \downarrow \} \).
  \[ \Rightarrow A \not\preceq_T \mathbb{H}_A. \]
A Generalization of the Friedberg-Muchnik Theorem

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

- $A$ is effectively enumerable by an machine in $M_A$.

  $\Rightarrow A \preceq_1 H_A$.

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

- Conditions for lowness for all $n > 0$:
  $(N_n)$ If $M_n^A(t)(n) \downarrow^t$ for infinitely many $t$, then $M_n^A(n) \downarrow$.

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

  $\Rightarrow A^C \not\preceq_1 H_A$.

- $K^A \preceq_T K^0$ where $K^O = \{k_M \mid M \in M^1_A(O) \& M(k_M) \downarrow\}$.

  $\Rightarrow A \not\preceq_T H_A$. 


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

- $A$ is effectively enumerable by an machine in $M_A$.
  \[ \Rightarrow A \leq_1 H_A. \]

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

- Conditions for lowness for all $n > 0$:
  \[
  (N_n) \text{ If } M^A_n (n) \downarrow^t \text{ for infinitely many } t, \text{ then } M^A_n (n) \downarrow.
  \]
  \[ \Rightarrow \text{ Conditions for simplicity for all } n > 0: \]
  \[
  (P_n) \text{ If } W_n = \bigcup_{i \geq 1} W_{n,i} \text{ is infinite, then } A \cap W_n \neq \emptyset.
  \]
  \[ \Rightarrow A^C \not\leq_1 H_A. \]

- $K^A \leq_T K^\emptyset$ where $K^\emptyset = \{ k_M | M \in M^1_A (O) \text{ & } M(k_M) \downarrow \}$.
  \[ \Rightarrow A \not\leq_T H_A. \]
The properties of $\mathbb{A}$, for instance, for $\mathcal{A} = (\mathbb{R}; 0, 1; +, −, \cdots; \leq)$:

- $\mathbb{A}$ is effectively enumerable by a machine in $M_\mathcal{A}$.

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

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

Conditions for lowness for all $n > 0$:

$(N_n)$ If $M^{\mathbb{A}_t}_n(n) \downarrow^t$ for infinitely many $t$, then $M^n_\mathcal{A}(n) \downarrow$.

$\implies$ Conditions for simplicity for all $n > 0$:

$(P_n)$ If $W_n = \bigcup_{i \geq 1} W_{n,i}$ is infinite, then $\mathbb{A} \cap W_n \neq \emptyset$.

$\implies \mathbb{A}^C \npreceq_1 \mathbb{H}_\mathcal{A}$.

- $K^{\mathbb{A}} \preceq_T K^0$ where $K^O = \{ k_M \mid M \in M^{1}_{\mathcal{A}}(O) \ \& \ M(k_M) \downarrow \}$.

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

- $\mathbb{A}$ is effectively enumerable by a machine in $M_{\mathcal{A}}$.
  \[ \Rightarrow \mathbb{A} \preceq_1 \mathbb{H}_{\mathcal{A}}. \]

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

- Conditions for lowness for all $n > 0$:
  - $(N_n)$ If $M_{\mathbb{A}^t}(n) \downarrow^t$ for infinitely many $t$, then $M_{\mathbb{A}}(n) \downarrow$.
  \[ \Rightarrow \text{Conditions for simplicity for all } n > 0: \]
  - $(P_n)$ If $W_n = \bigcup_{i \geq 1} W_{n,i}$ is infinite, then $\mathbb{A} \cap W_n \neq \emptyset$.
  \[ \Rightarrow \mathbb{A}^C \npreceq_1 \mathbb{H}_{\mathcal{A}}. \]

- $K^\mathbb{A} \preceq_T K^0$ where $K^O = \{ k_M \mid M \in M_{\mathcal{A}}(O) \& M(k_M) \downarrow \}$.
  \[ \Rightarrow \mathbb{A} \npreceq_T \mathbb{H}_{\mathcal{A}}. \]
The properties of $\mathbb{A}$, for instance, for $\mathcal{A} = (\mathbb{R}; 0, 1; +, -; \leq)$:

- $\mathbb{A}$ is effectively enumerable by a machine in $M_\mathcal{A}$.
  \[ \Rightarrow \mathbb{A} \preceq_1 \mathbb{H}_\mathcal{A} \]

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

- Conditions for lowness for all $n > 0$:
  $(N_n)$ If $M_{n}^{\mathbb{A}_t}(n) \downarrow^t$ for infinitely many $t$, then $M_{n}^{\mathbb{A}}(n) \downarrow$.
  \[ \Rightarrow \text{Conditions for simplicity for all } n > 0: \]
  $(P_n)$ If $W_n = \bigcup_{i \geq 1} W_{n,i}$ is infinite, then $\mathbb{A} \cap W_n \neq \emptyset$.
  \[ \Rightarrow \mathbb{A}^C \npreceq_1 \mathbb{H}_\mathcal{A} \]

- $K^{\mathbb{A}} \preceq_T K^\emptyset$ where $K^O = \{ k_M \mid M \in M_\mathcal{A}(O) \land M(k_M) \downarrow \}$.
  \[ \Rightarrow \mathbb{A} \npreceq_T \mathbb{H}_\mathcal{A} \]
Let $A = (\mathbb{R}; 0, 1; +, −, ⋅; ≤)$ or $A = (\mathbb{C}; 0, 1; +, −, ⋅; =)$ and
$P = A_{\text{Alg}},$
$P = \bigcup_{n \geq 1} \{ \vec{x} \in \mathbb{R}^n | (\exists \vec{q} \in \mathbb{Q}^n) (q_1 + \sum_{i=1}^{n-1} q_{i-1} x_i = x_n) \}$
$\mathbb{Z} \subseteq P$ or $\mathbb{Z} \cap P = \emptyset.$

**Lemma**
$A \nleq_T P.$

**Corollary**
$\mathbb{H}_A \nleq_T P.$

**Remark**
Similar constructions are also possible, if all problems which are semi-decidable by Turing machines are decidable over $A.$

$P \nleq_T \mathbb{H}_A$ holds for $A = (\mathbb{R}; 0, 1; +, −, \chi_{\text{TM}}, \phi; \leq)$ with $\phi(x) = \pi$
and $\pi \mathbb{Z} \subseteq P$ (where $A \subseteq \pi \mathbb{Z}$) and so on.
Let $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ or $\mathcal{A} = (\mathbb{C}; 0, 1; +, -, \cdot; =)$ and

$P = \mathcal{A}_{\text{Alg}}$

$P = \bigcup_{n \geq 1} \{ \vec{x} \in \mathbb{R}^n \mid (\exists \vec{q} \in \mathbb{Q}^n)(q_1 + \sum_{i=1}^{n-1} q_i - x_i = x_n) \}$,

$\mathbb{Z} \subseteq P$ or $\mathbb{Z} \cap P = \emptyset$.

**Lemma**

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

**Corollary**

$\mathbb{H}_A \not\preceq_T P$.  

**Remark**

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

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

Let $A = (\mathbb{R}; 0, 1; +, −, \cdot; \leq)$ or $A = (\mathbb{C}; 0, 1; +, −, \cdot; =)$ and

$P = A_{\mathrm{Alg}},$

$P = \bigcup_{n \geq 1} \{ \vec{x} \in \mathbb{R}^n \mid (\exists \vec{q} \in \mathbb{Q}^n)(q_1 + \sum_{i=1}^{n-1} q_{i-1}x_i = x_n) \},$

$\mathbb{Z} \subseteq P$ or $\mathbb{Z} \cap P = \emptyset.$

Lemma

$A \not\leq_T P.$

Corollary

$\mathbb{H}A \not\leq_T P.$

Remark

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

$P \not\leq_T \mathbb{H}A$ holds for $A = (\mathbb{R}; 0, 1; +, −, \chi_{\mathrm{TM}}, \phi; \leq)$ with $\phi(x) = \pi$ and $\pi\mathbb{Z} \subseteq P$ (where $A \subseteq \pi\mathbb{Z}$) and so on.
Let $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \cdot; \leq)$ or $\mathcal{A} = (\mathbb{C}; 0, 1; +, -, \cdot; =)$ and $P = \mathcal{A}_{\text{Alg}}$, $P = \bigcup_{n \geq 1} \{ \vec{x} \in \mathbb{R}^n \mid (\exists \vec{q} \in \mathbb{Q}^n)(q_1 + \sum_{i=1}^{n-1} q_{i-1}x_i = x_n) \}$, $\mathbb{Z} \subseteq P$ or $\mathbb{Z} \cap P = \emptyset$.

**Lemma**

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

**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 $\mathcal{A} = (\mathbb{R}; 0, 1; +, -, \chi_{\text{TM}}, \phi; \leq)$ with $\phi(x) = \pi$ and $\pi \mathbb{Z} \subseteq P$ (where $\mathcal{A} \subseteq \pi \mathbb{Z}$) and so on.
A Generalization of the Friedberg-Muchnik Theorem

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

Lemma
\[\mathbb{A} \not\leq_T P.\]

Corollary
\[\mathbb{H}_A \not\leq_T P.\]

Remark

Similar constructions are also possible, if all problems which are semi-decidable by Turing machines are decidable over \( A \).

\[ P \not\leq_T \mathbb{H}_A \text{ holds for } A = (\mathbb{R}; 0, 1; +, -, \chi_{\text{TM}}, \phi; \leq) \text{ with } \phi(x) = \pi \text{ and } \pi\mathbb{Z} \subseteq P \text{ (where } \mathbb{A} \subseteq \pi\mathbb{Z}) \text{ and so on.} \]
Lemma

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

Proof: Let us assume that \( A \) is decidable by a machine in \( M_A(P) \).

- \( \rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \) is semi-decidable by an \( M \in M_A(P) \).
- \( M \) can be modified:
  - The integers are enumerated and compared with the input.
  - If the input is a positive integer, then \( M \) can be simulated by a machine in \( M_A \) since
    - all queries of \( M \) are answered in the positive,
    - each order test can be simulated by means of equality tests.

- \( \rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \) is semi-decidable by a machine in \( M_A \).
- \( \rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} = W_j \) for some \( j \).
- \( \Rightarrow \) By definition of \( A \) the assumption is wrong.
Lemma

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

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

1. $\Rightarrow (\mathbb{R} \setminus \mathcal{A}) \cap \mathbb{N}$ is semi-decidable by an $\mathcal{M} \in M_{\mathcal{A}}(P)$.

2. $\mathcal{M}$ 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 $M_{\mathcal{A}}$ since all queries of $\mathcal{M}$ are answered in the positive, each order test can be simulated by means of equality tests.

3. $\Rightarrow (\mathbb{R} \setminus \mathcal{A}) \cap \mathbb{N}$ is semi-decidable by a machine in $M_{\mathcal{A}}$.

4. $\Rightarrow (\mathbb{R} \setminus \mathcal{A}) \cap \mathbb{N} = W_j$ for some $j$.

5. $\Rightarrow$ By definition of $\mathcal{A}$ the assumption is wrong.
Lemma

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

Proof: Let us assume that \( A \) is decidable by a machine in \( M_A(P) \).

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \text{ is semi-decidable by an } M \in M_A(P). \]

\( M \) can be modified:

- The integers are enumerated and compared with the input.
- If the input is a positive integer, then \( M \) can be simulated by a machine in \( M_A \) since
  - all queries of \( M \) are answered in the positive,
  - each order test can be simulated by means of equality tests.

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \text{ is semi-decidable by a machine in } M_A. \]

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} = W_j \text{ for some } j. \]

\[ \Rightarrow \text{By definition of } A \text{ the assumption is wrong.} \]
Lemma

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

Proof: Let us assume that \( A \) is decidable by a machine in \( M_A(P) \).

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \] is semi-decidable by an \( M \in M_A(P) \).

\( M \) can be modified:

- The integers are enumerated and compared with the input.
- If the input is a positive integer,
  then \( M \) can be simulated by a machine in \( M_A \) since
    - all queries of \( M \) are answered in the positive,
    - each order test can be simulated by means of equality tests.

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \] is semi-decidable by a machine in \( M_A \).

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} = W_j \text{ for some } j. \]

\[ \Rightarrow \text{By definition of } A \text{ the assumption is wrong}. \]
Lemma

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

Proof: Let us assume that \( A \) is decidable by a machine in \( M_A(P) \).

- \( \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \) is semi-decidable by an \( \mathcal{M} \in M_A(P) \).
- \( \mathcal{M} \) 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 \( M_A \) since all queries of \( \mathcal{M} \) are answered in the positive, each order test can be simulated by means of equality tests.
  - \( \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \) is semi-decidable by a machine in \( M_A \).
  - \( \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} = W_j \) for some \( j \).
  - \( \Rightarrow \) By definition of \( A \) the assumption is wrong.
Lemma

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

Proof: Let us assume that $A$ is decidable by a machine in $M_A(P)$.

$\Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N}$ is semi-decidable by an $M \in M_A(P)$.

$M$ can be modified:

- The integers are enumerated and compared with the input.
- If the input is a positive integer, then $M$ can be simulated by a machine in $M_A$ since
  - all queries of $M$ are answered in the positive,
  - each order test can be simulated by means of equality tests.

$\Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N}$ is semi-decidable by a machine in $M_A$.

$\Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} = W_j$ for some $j$.

$\Rightarrow$ By definition of $A$ the assumption is wrong.
Lemma

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

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

- \( \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \) is semi-decidable by an \( \mathcal{M} \in \mathcal{M}_A(P) \).
- \( \mathcal{M} \) 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 \( \mathcal{M}_A \) since
    - all queries of \( \mathcal{M} \) are answered in the positive,
    - each order test can be simulated by means of equality tests.

- \( \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \) is semi-decidable by a machine in \( \mathcal{M}_A \).

- \( \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} = W_j \) for some \( j \).
- \( \Rightarrow \) By definition of \( A \) the assumption is wrong.
Lemma

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

Proof: Let us assume that \( A \) is decidable by a machine in \( M_A(P) \).

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \text{ is semi-decidable by an } M \in M_A(P). \]

\( M \) can be modified:

- The integers are enumerated and compared with the input.
- If the input is a positive integer, then \( M \) can be simulated by a machine in \( M_A \) since
  - all queries of \( M \) are answered in the positive,
  - each order test can be simulated by means of equality tests.

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \text{ is semi-decidable by a machine in } M_A. \]

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} = W_j \text{ for some } j. \]

\[ \Rightarrow \text{By definition of } A \text{ the assumption is wrong.} \]
A Generalization of the Friedberg-Muchnik Theorem

Lemma

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

Proof: Let us assume that \( A \) is decidable by a machine in \( M_A(P) \).

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \text{ is semi-decidable by an } M \in M_A(P). \]
\[ M \text{ can be modified:} \]

- The integers are enumerated and compared with the input.
- If the input is a positive integer, then \( M \) can be simulated by a machine in \( M_A \) since
  - all queries of \( M \) are answered in the positive,
  - each order test can be simulated by means of equality tests.

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \text{ is semi-decidable by a machine in } M_A. \]

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} = W_j \text{ for some } j. \]

\[ \Rightarrow \text{ By definition of } A \text{ the assumption is wrong.} \]
A Generalization of the Friedberg-Muchnik Theorem

Lemma

\[ A \not\preceq_T P \text{ for } A = (\mathbb{R}; 0, 1; +, -, \cdot; \leq) \text{ and } \mathbb{Z} \subseteq P \subseteq U. \]

Proof: Let us assume that \( A \) is decidable by a machine in \( M_A(P) \).

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \text{ is semi-decidable by an } M \in M_A(P). \]

\( M \) can be modified:

- The integers are enumerated and compared with the input.
- If the input is a positive integer, then \( M \) can be simulated by a machine in \( M_A \) since all queries of \( M \) are answered in the positive, each order test can be simulated by means of equality tests.

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} \text{ is semi-decidable by a machine in } M_A. \]

\[ \Rightarrow (\mathbb{R} \setminus A) \cap \mathbb{N} = W_j \text{ for some } j. \]

\[ \Rightarrow \text{ By definition of } A \text{ the assumption is wrong.} \]
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 $E = \text{df} \bigcup_{i \geq 0} E_i$.

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 $E^\infty$.

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