Reverse Mathematics & Nonstandard Analysis:

Making sense of infinite computations.

Sam Sanders[1]

Tohoku University & Ghent University

CiE, June 29, 2011, Sofia

[1]This research is generously supported by the John Templeton Foundation.

Goal & Motivation

Goal & Motivation

Goal: To formalize Computability using Nonstandard Analysis.

Goal & Motivation

Goal: To formalize Computability using Nonstandard Analysis.

Motivation:

Goal & Motivation

Goal: To formalize Computability using Nonstandard Analysis.

Motivation: Erret Bishop (and others) have derided NSA for its ‘lack of computational content’.

Nonstandard Analysis

Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

Nonstandard Analysis

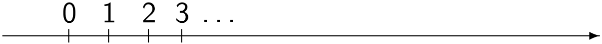
Nonstandard Analysis formalizes 'calculus with infinitesimals'.

For this talk, we only need ${}^*\mathbb{N}$.

Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

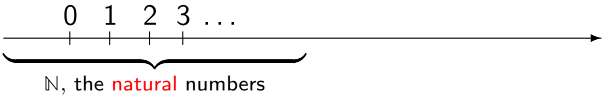
For this talk, we only need ${}^*\mathbb{N}$.



Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

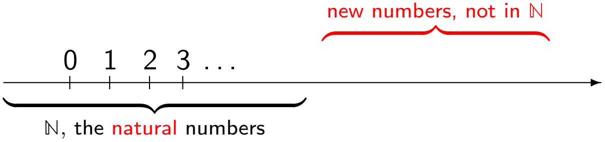
For this talk, we only need $^*\mathbb{N}$.



Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

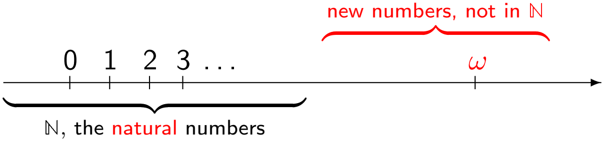
For this talk, we only need ${}^*\mathbb{N}$.



Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

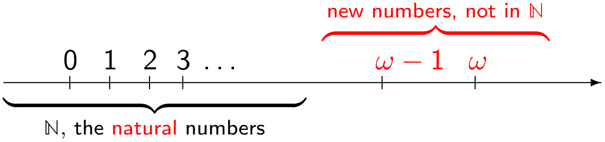
For this talk, we only need ${}^*\mathbb{N}$.



Nonstandard Analysis

Nonstandard Analysis formalizes 'calculus with infinitesimals'.

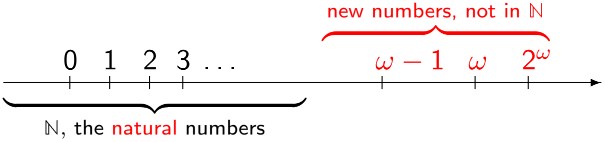
For this talk, we only need $^*\mathbb{N}$.



Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

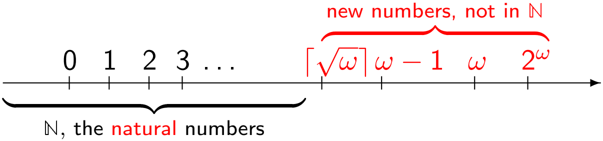
For this talk, we only need $^*\mathbb{N}$.



Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

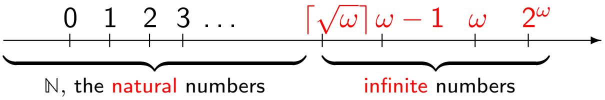
For this talk, we only need ${}^*\mathbb{N}$.



Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

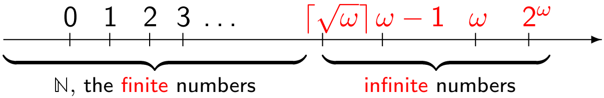
For this talk, we only need ${}^*\mathbb{N}$.



Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

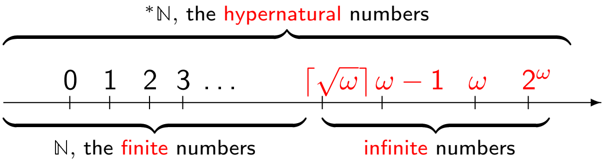
For this talk, we only need $^*\mathbb{N}$.



Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

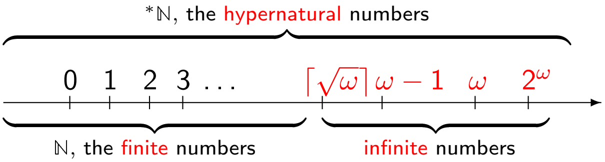
For this talk, we only need $^*\mathbb{N}$.



Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

For this talk, we only need ${}^*\mathbb{N}$.

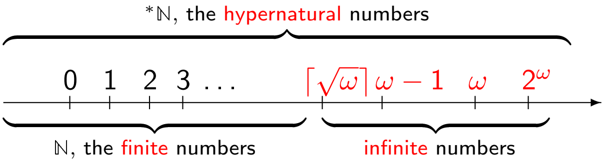


$\mathbb{N}=$ finite (or standard) numbers

Nonstandard Analysis

Nonstandard Analysis formalizes ‘calculus with infinitesimals’.

For this talk, we only need ${}^*\mathbb{N}$.



$\mathbb{N}=$ finite (or standard) numbers

${}^*\mathbb{N}\setminus\mathbb{N}=$ infinite (or nonstandard) numbers

A simple example

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) Define $m := 0$.

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) Define $m := 0$**.**
1) Check $\varphi(m)$**.**

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) Define $m := 0$**.**
1) Check $\varphi(m)$**.**
2) If $\varphi(m)$ **is TRUE, return** m**,**

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) Define $m := 0$**.**
1) Check $\varphi(m)$**.**
2) If $\varphi(m)$ **is TRUE, return** m**, otherwise define** $m := m + 1$ **and go to 1).**

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) Define $m := 0$.
1) Check $\varphi(m)$.
2) If $\varphi(m)$ is TRUE, return m, otherwise define $m := m + 1$ and go to 1).

The program $\mathbb{A}$ will certainly halt.

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) Define $m := 0$.
1) Check $\varphi(m)$.
2) If $\varphi(m)$ is TRUE, return m, otherwise define $m := m + 1$ and go to 1).

The program $\mathbb{A}$ will certainly halt.

Nonstandard program $\mathbb{B}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) Define $m := 0$.
1) Check $\varphi(m)$.
2) If $\varphi(m)$ is TRUE, return m, otherwise define $m := m + 1$ and go to 1).

The program $\mathbb{A}$ will certainly halt.

Nonstandard program $\mathbb{B}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) For $i = 0.\ldots.\omega$ do

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) Define $m := 0$.
1) Check $\varphi(m)$.
2) If $\varphi(m)$ is TRUE, return m, otherwise define $m := m + 1$ and go to 1).

The program $\mathbb{A}$ will certainly halt.

Nonstandard program $\mathbb{B}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) For $i = 0 \ldots \ldots \omega$ do If $\varphi(i)$ is TRUE, return i and halt.

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) Define $m := 0$**.**
1) Check $\varphi(m)$**.**
2) If $\varphi(m)$ **is TRUE, return** m**, otherwise define** $m := m + 1$ **and go to 1).**

The program $\mathbb{A}$ will certainly halt.

Nonstandard program $\mathbb{B}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) For $i = 0, \dots, \omega$ **do If** $\varphi(i)$ **is TRUE, return** i **and halt. Continue otherwise.**

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) Define $m := 0$**.**
1) Check $\varphi(m)$**.**
2) If $\varphi(m)$ **is TRUE, return** m**, otherwise define** $m := m + 1$ **and go to 1).**

The program $\mathbb{A}$ will certainly halt.

Nonstandard program $\mathbb{B}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) For $i = 0.\dots.\omega$ **do If** $\varphi(i)$ **is TRUE, return** i **and halt. Continue otherwise.**

The program $\mathbb{B}$ will certainly halt at some finite stage.

A simple example

Suppose we have proved $(\exists n \in \mathbb{N})\varphi(n)$. ($\varphi$ is quantifier-free)

Program $\mathbb{A}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) Define $m := 0$.
1) Check $\varphi(m)$.
2) If $\varphi(m)$ is TRUE, return m, otherwise define $m := m + 1$ and go to 1).

The program $\mathbb{A}$ will certainly halt.

Nonstandard program $\mathbb{B}$ to find $n_0 \in \mathbb{N}$ s.t. $\varphi(n_0)$:

0) For $i = 0, \dots, \omega$ do If $\varphi(i)$ is TRUE, return i and halt. Continue otherwise.

The program $\mathbb{B}$ will certainly halt at some finite stage.

$\mathbb{B}$ depends on the infinite ω, but not on the choice of ω.

Turing Computability

We always assume that $A \subset \mathbb{N} \subset {}^*\mathbb{N}$ and that ω is infinite.

Turing Computability

We always assume that $A \subset \mathbb{N} \subset {}^*\mathbb{N}$ and that ω is infinite.

Definition

The set A is ω-invariant if there is $\psi \in \Delta_0$ s.t. for all infinite ω ,

Turing Computability

We always assume that $A \subset \mathbb{N} \subset {}^*\mathbb{N}$ and that ω is infinite.

Definition

The set A is ω-invariant if there is $\psi \in \Delta_0$ s.t. for all infinite ω ,

$$A = \{k \in \mathbb{N} : \psi(k, \omega)\}.$$

Turing Computability

We always assume that $A \subset \mathbb{N} \subset {}^*\mathbb{N}$ and that ω is infinite.

Definition

The set A is ω-invariant if there is $\psi \in \Delta_0$ s.t. for all infinite ω ,

$$A = \{k \in \mathbb{N} : \psi(k, \omega)\}.$$

The set A depends on ω, but not on the choice of ω.

Turing Computability

We always assume that $A \subset \mathbb{N} \subset {}^*\mathbb{N}$ and that ω is infinite.

Definition

The set A is ω-invariant if there is $\psi \in \Delta_0$ s.t. for all infinite ω ,

$$A = \{k \in \mathbb{N} : \psi(k, \omega)\}.$$

The set A depends on ω, but not on the choice of ω.

Theorem

The Δ_1-sets (=Turing computable) are exactly the ω-invariant sets.

The Limit Lemma

The Limit Lemma

Theorem (Limit lemma)

$f \leq_T \mathbf{0}' \iff f \in \Delta_2 \iff f = \lim_{n\to\infty} f_n$ (f_n is computable)

The Limit Lemma

Theorem (Limit lemma)

$f \leq_T \mathbf{0}' \iff f \in \Delta_2 \iff f = \lim_{n\to\infty} f_n \qquad (f_n \text{ is computable})$

$\mathbf{0}'$ is a decision procedure for Σ_1-formulas called ‘halting problem’.

The Limit Lemma

Theorem (Limit lemma)

$f \leq_T \mathbf{0}' \iff f \in \Delta_2 \iff f = \lim_{n\to\infty} f_n$ (f_n is computable)

$\mathbf{0}'$ is a decision procedure for Σ_1-formulas called 'halting problem'.

Theorem (Hyperlimit Lemma)

$f \leq_T \mathbf{\Pi_1} \iff f \in \Delta_2 \iff f = f_\omega$ (f_n is computable)

The Limit Lemma

Theorem (Limit lemma)

$f \leq_T \mathbf{0}' \iff f \in \Delta_2 \iff f = \lim_{n\to\infty} f_n$ (f_n is computable)

$\mathbf{0}'$ is a decision procedure for Σ_1-formulas called ‘halting problem’.

Theorem (Hyperlimit Lemma)

$f \leq_T \mathbf{\Pi_1} \iff f \in \Delta_2 \iff f = f_\omega$ (f_n is computable)

$\mathbf{\Pi_1}$ is a decision procedure for Σ_1-formulas, given by:

The Limit Lemma

Theorem (Limit lemma)

$f \leq_T \mathbf{0}' \iff f \in \Delta_2 \iff f = \lim_{n\to\infty} f_n \qquad (f_n \text{ is computable})$

$\mathbf{0}'$ is a decision procedure for Σ_1-formulas called 'halting problem'.

Theorem (Hyperlimit Lemma)

$f \leq_T \mathbf{\Pi_1} \iff f \in \Delta_2 \iff f = f_\omega \qquad (f_n \text{ is computable})$

$\mathbf{\Pi_1}$ is a decision procedure for Σ_1-formulas, given by:

Theorem ($\mathbf{\Pi_1}$)

For every $\varphi \in \Delta_0$, we have $(\forall n \in \mathbb{N})\varphi(n) \rightarrow (\forall n \in {}^*\mathbb{N})\varphi(n)$.

The Limit Lemma

Theorem (Limit lemma)

$f \leq_T \mathbf{0'} \iff f \in \Delta_2 \iff f = \lim_{n\to\infty} f_n$ (f_n is computable)

$\mathbf{0'}$ is a decision procedure for Σ_1-formulas called 'halting problem'.

Theorem (Hyperlimit Lemma)

$f \leq_T \mathbf{\Pi_1} \iff f \in \Delta_2 \iff f = f_\omega$ (f_n is computable)

$\mathbf{\Pi_1}$ is a decision procedure for Σ_1-formulas, given by:

Theorem ($\mathbf{\Pi_1}$)

For every $\varphi \in \Delta_0$, we have $(\forall n \in \mathbb{N})\varphi(n) \to (\forall n \in {}^*\mathbb{N})\varphi(n)$.

Also called 'Transfer principle for Π_1-formulas' or 'Π_1-transfer'.

Origins in RM

Origins in RM

ERNA = Nonstandard Analysis in $I\Delta_0 + \exp$.

Origins in RM

ERNA = Nonstandard Analysis in $I\Delta_0 + \exp$.

RCA_0 defines exactly the Δ_1-sets.

Origins in RM

ERNA = Nonstandard Analysis in $I\Delta_0 + \exp$.

RCA_0 defines exactly the Δ_1-sets.

"Theorem"

Origins in RM

ERNA = Nonstandard Analysis in $I\Delta_0 + \exp$.

RCA_0 defines exactly the Δ_1-sets.

"Theorem"

1. *If* RCA_0 *proves* $T(=)$, *then* ERNA *proves* $T(\approx)$.

Origins in RM

ERNA = Nonstandard Analysis in $I\Delta_0 + \exp$.

RCA_0 defines exactly the Δ_1-sets.

"Theorem"

1. *If* RCA_0 *proves* $T(=)$, *then* ERNA *proves* $T(\approx)$.
2. *If* RCA_0 *proves* $[T(=) \Leftrightarrow \mathsf{WKL}_0]$,

Origins in RM

ERNA = Nonstandard Analysis in $I\Delta_0 + \exp$.

RCA_0 defines exactly the Δ_1-sets.

“Theorem”

1. *If* RCA_0 *proves* $T(=)$, *then* ERNA *proves* $T(\approx)$.
2. *If* RCA_0 *proves* $[T(=) \Leftrightarrow \mathsf{WKL}_0]$, *then* ERNA *proves* $[T(\approx) \Leftrightarrow \Pi_1\text{-TRANS}]$.

Origins in RM

ERNA = Nonstandard Analysis in $I\Delta_0 + \exp$.

RCA_0 defines exactly the Δ_1-sets.

“Theorem”

1. *If* RCA_0 *proves* $T(=)$, *then* ERNA *proves* $T(\approx)$.
2. *If* RCA_0 *proves* $[T(=) \Leftrightarrow \mathrm{WKL}_0]$, *then* ERNA *proves* $[T(\approx) \Leftrightarrow \Pi_1\text{-TRANS}]$.

Here, $T(=)$ is a theorem of ordinary Mathematics.

Origins in RM

ERNA = Nonstandard Analysis in $I\Delta_0 + \exp$.

RCA_0 defines exactly the Δ_1-sets.

"Theorem"

1. *If* RCA_0 *proves* $T(=)$, *then* ERNA *proves* $T(\approx)$.
2. *If* RCA_0 *proves* $[T(=) \Leftrightarrow WKL_0]$, *then* ERNA *proves* $[T(\approx) \Leftrightarrow \Pi_1\text{-TRANS}]$.

Here, $T(=)$ is a theorem of ordinary Mathematics.

Example of 1: Intermediate Value Theorem.

Origins in RM

ERNA = Nonstandard Analysis in $I\Delta_0 + \exp$.

RCA_0 defines exactly the Δ_1-sets.

"Theorem"

1. *If* RCA_0 *proves* $T(=)$, *then* ERNA *proves* $T(\approx)$.
2. *If* RCA_0 *proves* $[T(=) \Leftrightarrow WKL_0]$, *then* ERNA *proves* $[T(\approx) \Leftrightarrow \Pi_1\text{-TRANS}]$.

Here, $T(=)$ is a theorem of ordinary Mathematics.

Example of 1: Intermediate Value Theorem.
Example of 2: Peano's theorem for diff. eq. $y' = f(x, y)$.

Origins in RM

ERNA = Nonstandard Analysis in $I\Delta_0 + \exp$.

RCA_0 defines exactly the Δ_1-sets.

"Theorem"

1. *If* RCA_0 *proves* $T(=)$, *then* ERNA *proves* $T(\approx)$.
2. *If* RCA_0 *proves* $[T(=) \Leftrightarrow WKL_0]$, *then* ERNA *proves* $[T(\approx) \Leftrightarrow \Pi_1\text{-TRANS}]$.

Here, $T(=)$ is a theorem of ordinary Mathematics.

Example of 1: Intermediate Value Theorem.
Example of 2: Peano's theorem for diff. eq. $y' = f(x, y)$.

But RCA_0 and WKL_0 are recursion theoretic!

Origins in RM

ERNA = Nonstandard Analysis in $I\Delta_0 + \exp$.

RCA_0 defines exactly the Δ_1-sets.

"Theorem"

1. *If* RCA_0 *proves* $T(=)$, *then* ERNA *proves* $T(\approx)$.
2. *If* RCA_0 *proves* $[T(=) \Leftrightarrow WKL_0]$, *then* ERNA *proves* $[T(\approx) \Leftrightarrow \Pi_1\text{-TRANS}]$.

Here, $T(=)$ is a theorem of ordinary Mathematics.

Example of 1: Intermediate Value Theorem.
Example of 2: Peano's theorem for diff. eq. $y' = f(x, y)$.

But RCA_0 and WKL_0 are recursion theoretic!

How about ERNA and Π_1-TRANS?

Constructive Reverse Mathematics

CRM = RM in Bishop's 'constructive analysis'.

Constructive Reverse Mathematics

CRM = RM in Bishop's 'constructive analysis'.

An important principle is:

Principle (Σ_1-excluded middle or LPO)

For every q.f. formula φ, we have $(\exists n \in \mathbb{N})\varphi(n) \vee (\forall n \in \mathbb{N})\neg\varphi(n)$.

Constructive Reverse Mathematics

CRM = RM in Bishop's 'constructive analysis'.

An important principle is:

Principle (Σ_1-excluded middle or LPO)

For every q.f. formula φ, we have $(\exists n \in \mathbb{N})\varphi(n) \vee (\forall n \in \mathbb{N})\neg\varphi(n)$.

The previous principle states: There is a finite procedure that decides whether $(\exists n \in \mathbb{N})\varphi(n)$ or not.

Constructive Reverse Mathematics

CRM = RM in Bishop's 'constructive analysis'.

An important principle is:

Principle (Σ_1-excluded middle or LPO)

For every q.f. formula φ, we have $(\exists n \in \mathbb{N})\varphi(n) \vee (\forall n \in \mathbb{N})\neg\varphi(n)$.

The previous principle states: There is a finite procedure that decides whether $(\exists n \in \mathbb{N})\varphi(n)$ or not.

Principle ($\mathbf{\Pi_1}$-Transfer)

For every q.f. formula φ, we have $(\exists n \in \mathbb{N})\varphi(n) \vee (\forall n \in {}^*\mathbb{N})\neg\varphi(n)$.

Constructive Reverse Mathematics

CRM = RM in Bishop's 'constructive analysis'.

An important principle is:

Principle (Σ_1-excluded middle or LPO)

For every q.f. formula φ, we have $(\exists n \in \mathbb{N})\varphi(n) \vee (\forall n \in \mathbb{N})\neg\varphi(n)$.

The previous principle states: There is a finite procedure that decides whether $(\exists n \in \mathbb{N})\varphi(n)$ or not.

Principle ($\mathbf{\Pi}_1$-Transfer)

For every q.f. formula φ, we have $(\exists n \in \mathbb{N})\varphi(n) \vee (\forall n \in {}^*\mathbb{N})\neg\varphi(n)$.

The previous principle is equivalent to: There is an ω-invariant procedure that decides whether $(\exists n \in \mathbb{N})\varphi(n)$ or not.

Constructive Reverse Mathematics

Constructive Reverse Mathematics

In CRM, LPO is equivalent to MCT and to

Constructive Reverse Mathematics

In CRM, LPO is equivalent to MCT and to

Principle

$(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$

Constructive Reverse Mathematics

In CRM, LPO is equivalent to MCT and to

Principle

$(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$

The previous principle should be read: For $x \in \mathbb{R}$, there is a finite procedure that decides if $x > 0$.

Constructive Reverse Mathematics

In CRM, LPO is equivalent to MCT and to

Principle

$(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$

The previous principle should be read: For $x \in \mathbb{R}$, there is a finite procedure that decides if $x > 0$.

In NSA, Π_1-TRANS is equivalent to MCT($\approx$) and to

Constructive Reverse Mathematics

In CRM, LPO is equivalent to MCT and to

Principle

$(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$

The previous principle should be read: For $x \in \mathbb{R}$, there is a finite procedure that decides if $x > 0$.

In NSA, Π_1-TRANS is equivalent to MCT($\approx$) and to

Principle

For $x \in \mathbb{R}$, there is an ω-invariant procedure that decides if $x > 0$.

Lost in translation

Lost in translation

Hypernegation provides a translation between NSA and CRM:

Lost in translation

Hypernegation provides a translation between NSA and CRM:

Definition (Hypernegation)

- $\sim [(\exists n \in \mathbb{N})\varphi(n)]$

Lost in translation

Hypernegation provides a translation between NSA and CRM:

Definition (Hypernegation)

- $\sim[(\exists n \in \mathbb{N})\varphi(n)] \equiv (\forall n \in {}^*\mathbb{N})\neg\varphi(n)$.

Lost in translation

Hypernegation provides a translation between NSA and CRM:

Definition (Hypernegation)

- $\sim [(\exists n \in \mathbb{N})\varphi(n)] \equiv (\forall n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in \mathbb{N})\varphi(n)] \equiv (\exists n \in {}^*\mathbb{N})\neg\varphi(n)$.

Lost in translation

Hypernegation provides a translation between NSA and CRM:

Definition (Hypernegation)

- $\sim [(\exists n \in \mathbb{N})\varphi(n)] \equiv (\forall n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in \mathbb{N})\varphi(n)] \equiv (\exists n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in {}^*\mathbb{N})\varphi(n)]$

Lost in translation

Hypernegation provides a translation between NSA and CRM:

Definition (Hypernegation)

- $\sim [(\exists n \in \mathbb{N})\varphi(n)] \equiv (\forall n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in \mathbb{N})\varphi(n)] \equiv (\exists n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in {}^*\mathbb{N})\varphi(n)] \equiv (\exists n \leq \omega)\neg\varphi(n)$.

Lost in translation

Hypernegation provides a translation between NSA and CRM:

Definition (Hypernegation)

- $\sim [(\exists n \in \mathbb{N})\varphi(n)] \equiv (\forall n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in \mathbb{N})\varphi(n)] \equiv (\exists n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in {}^*\mathbb{N})\varphi(n)] \equiv (\exists n \leq \omega)\neg\varphi(n)$. ($\omega$ is independent of parameters in φ)

Lost in translation

Hypernegation provides a translation between NSA and CRM:

Definition (Hypernegation)

- $\sim [(\exists n \in \mathbb{N})\varphi(n)] \equiv (\forall n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in \mathbb{N})\varphi(n)] \equiv (\exists n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in {}^*\mathbb{N})\varphi(n)] \equiv (\exists n \leq \omega)\neg\varphi(n)$. ($\omega$ is independent of parameters in φ)
- $\sim [(\exists n \in {}^*\mathbb{N})\varphi(n)] \equiv (\forall n \leq \omega)\neg\varphi(n)$.

Lost in translation

Hypernegation provides a translation between NSA and CRM:

Definition (Hypernegation)

- $\sim [(\exists n \in \mathbb{N})\varphi(n)] \equiv (\forall n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in \mathbb{N})\varphi(n)] \equiv (\exists n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in {}^*\mathbb{N})\varphi(n)] \equiv (\exists n \leq \omega)\neg\varphi(n)$. ($\omega$ is independent of parameters in φ)
- $\sim [(\exists n \in {}^*\mathbb{N})\varphi(n)] \equiv (\forall n \leq \omega)\neg\varphi(n)$.

With the hypernegation $\sim$, we get the usual results from CRM:

Lost in translation

Hypernegation provides a translation between NSA and CRM:

Definition (Hypernegation)

- $\sim[(\exists n \in \mathbb{N})\varphi(n)] \equiv (\forall n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim[(\forall n \in \mathbb{N})\varphi(n)] \equiv (\exists n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim[(\forall n \in {}^*\mathbb{N})\varphi(n)] \equiv (\exists n \leq \omega)\neg\varphi(n)$. ($\omega$ is independent of parameters in φ)
- $\sim[(\exists n \in {}^*\mathbb{N})\varphi(n)] \equiv (\forall n \leq \omega)\neg\varphi(n)$.

With the hypernegation $\sim$, we get the usual results from CRM:

Theorem

In NSA, LPO is equivalent to MP plus LLPO

Lost in translation

Hypernegation provides a translation between NSA and CRM:

Definition (Hypernegation)

- $\sim [(\exists n \in \mathbb{N})\varphi(n)] \equiv (\forall n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in \mathbb{N})\varphi(n)] \equiv (\exists n \in {}^*\mathbb{N})\neg\varphi(n)$.
- $\sim [(\forall n \in {}^*\mathbb{N})\varphi(n)] \equiv (\exists n \leq \omega)\neg\varphi(n)$. ($\omega$ is independent of parameters in φ)
- $\sim [(\exists n \in {}^*\mathbb{N})\varphi(n)] \equiv (\forall n \leq \omega)\neg\varphi(n)$.

With the hypernegation $\sim$, we get the usual results from CRM:

Theorem

In NSA, LPO is equivalent to MP plus LLPO

LPO: $P \vee \sim P$, MP: $\sim\sim P \to P$, LLPO: $\sim(P \wedge Q) \to \sim P \vee \sim Q$ $(P, Q \in \Sigma_1)$

Lost in translation

Theorem

In NSA, TFAE

1. LLPO

Lost in translation

Theorem

In NSA, TFAE

1. LLPO
2. $(\forall x \in \mathbb{R})(\sim(x > 0) \vee \sim (x < 0))$

Lost in translation

Theorem

In NSA, TFAE

1. LLPO
2. $(\forall x \in \mathbb{R})(\sim(x > 0) \vee \sim (x < 0))$
3. $(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$

Lost in translation

Theorem

In NSA, TFAE

1. LLPO
2. $(\forall x \in \mathbb{R})(\sim(x > 0) \vee \sim (x < 0))$
3. $(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$

LLPO: $\sim(P \wedge Q) \rightarrow \sim P \vee \sim Q$ $(P, Q \in \Sigma_1)$

Lost in translation

Theorem

In NSA, TFAE

1. LLPO
2. $(\forall x \in \mathbb{R})(\sim(x > 0) \vee \sim (x < 0))$
3. $(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$

LLPO: $\sim(P \wedge Q) \rightarrow \sim P \vee \sim Q$ $(P, Q \in \Sigma_1)$

Why does this connection exist?

Lost in translation

Theorem

In NSA, TFAE

1. LLPO
2. $(\forall x \in \mathbb{R})(\sim(x > 0) \vee \sim (x < 0))$
3. $(\forall x, y \in \mathbb{R})(xy = 0 \to x = 0 \vee y = 0)$

LLPO: $\sim(P \wedge Q) \to \sim P \vee \sim Q$ $(P, Q \in \Sigma_1)$

Why does this connection exist?

Compare $\mathbb{N}$ and $\mathcal{N}$.

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

The nonstandard Turing hierarchy

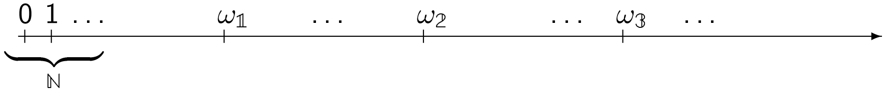
In classical NSA, a number is either finite or infinite.

In ‘stratified’ or ‘relative’ NSA, there are ‘levels’ of infinity.

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

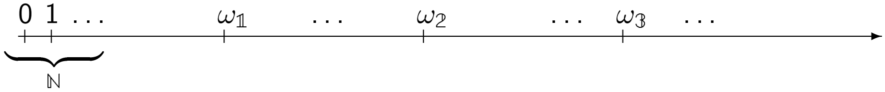
In 'stratified' or 'relative' NSA, there are 'levels' of infinity. Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:



The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In ‘stratified’ or ‘relative’ NSA, there are ‘levels’ of infinity. Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:

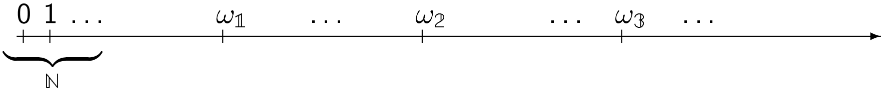


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \ldots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In 'stratified' or 'relative' NSA, there are 'levels' of infinity. Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:

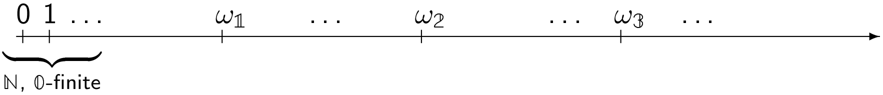


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \dots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In 'stratified' or 'relative' NSA, there are 'levels' of infinity.
Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:

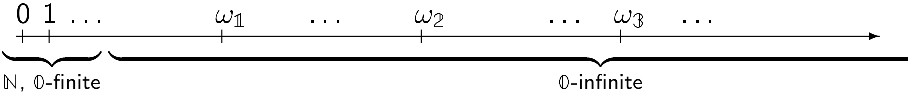


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \dots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In ‘stratified’ or ‘relative’ NSA, there are ‘levels’ of infinity.
Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:

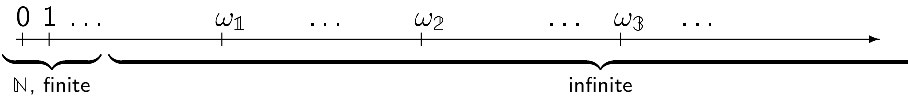


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \ldots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In ‘stratified’ or ‘relative’ NSA, there are ‘levels’ of infinity.
Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:

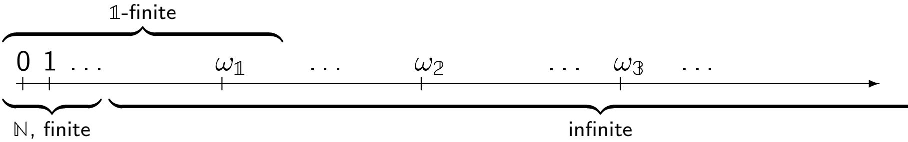


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \dots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In 'stratified' or 'relative' NSA, there are 'levels' of infinity.
Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:

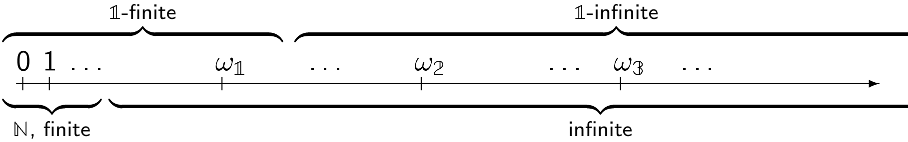


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \ldots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In 'stratified' or 'relative' NSA, there are 'levels' of infinity.
Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:

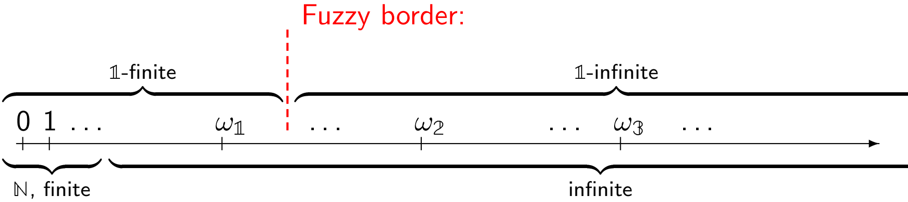


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \dots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In ‘stratified’ or ‘relative’ NSA, there are ‘levels’ of infinity.
Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:

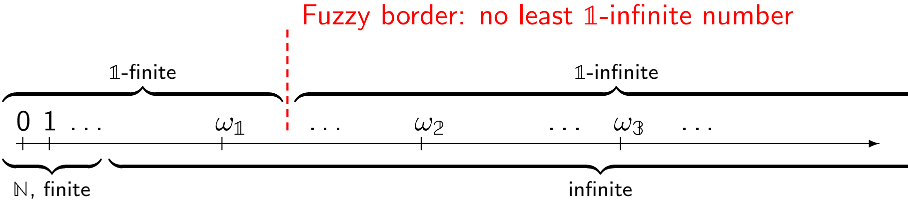


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \ldots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In 'stratified' or 'relative' NSA, there are 'levels' of infinity. Consider $\mathbb{N}^{\mathbb{A}}$, an extension of $^*\mathbb{N}$:

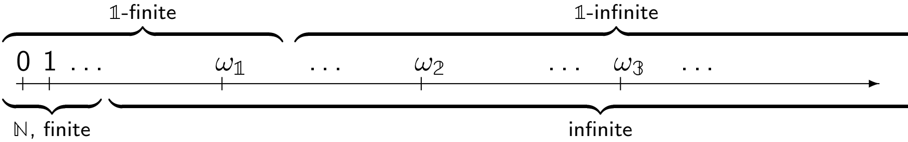


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \ldots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In 'stratified' or 'relative' NSA, there are 'levels' of infinity.
Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:

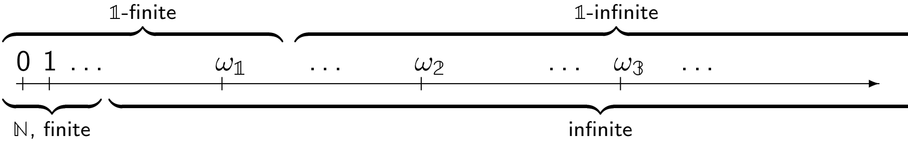


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \ldots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In ‘stratified’ or ‘relative’ NSA, there are ‘levels’ of infinity. Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:

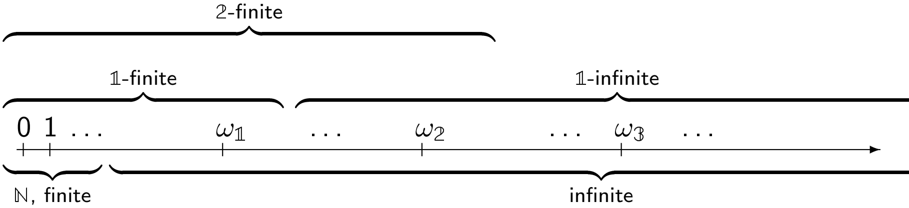


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \dots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In ‘stratified’ or ‘relative’ NSA, there are ‘levels’ of infinity.
Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:

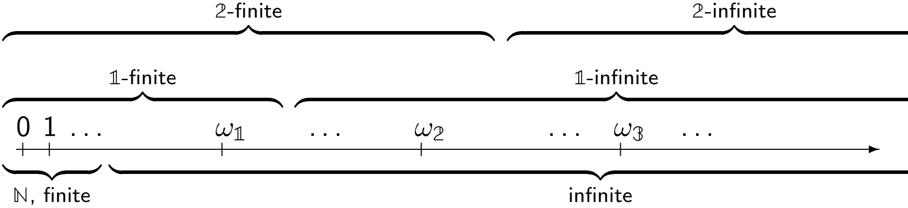


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \dots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In ‘stratified’ or ‘relative’ NSA, there are ‘levels’ of infinity.
Consider $\mathbb{N}^{\mathbb{A}}$, an extension of $^*\mathbb{N}$:

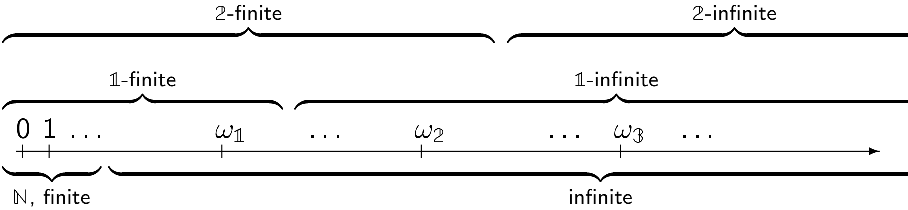


where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \dots\}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In 'stratified' or 'relative' NSA, there are 'levels' of infinity.
Consider $\mathbb{N}^{\mathbb{A}}$, an extension of $^*\mathbb{N}$:



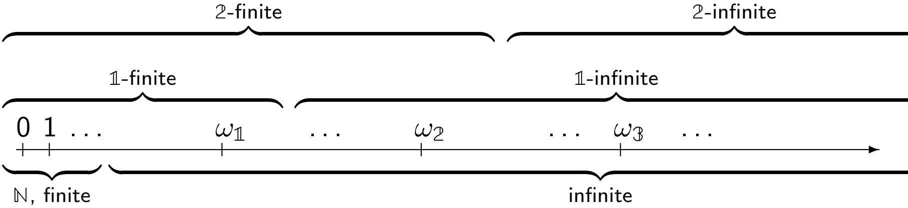
where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \dots\}$

The infinite numbers are 'stratified' in $|\mathbb{A}|$ many levels of infinity.

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In ‘stratified’ or ‘relative’ NSA, there are ‘levels’ of infinity. Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:



where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \dots\}$

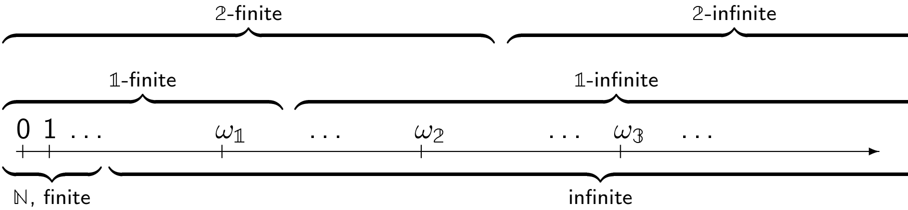
The infinite numbers are ‘stratified’ in $|\mathbb{A}|$ many levels of infinity.

Then $\omega_{\mathbb{2}}$ is infinite ‘relative’ to $\omega_{\mathbb{1}}$

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In 'stratified' or 'relative' NSA, there are 'levels' of infinity. Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:



where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \ldots\}$

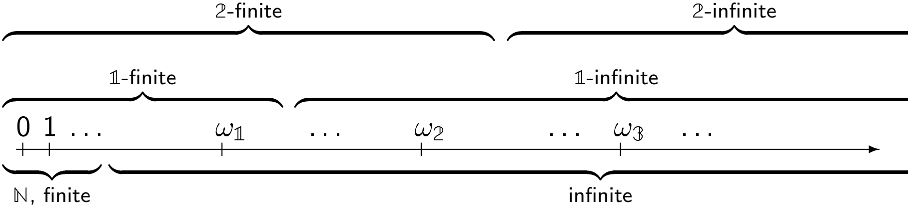
The infinite numbers are 'stratified' in $|\mathbb{A}|$ many levels of infinity.

Then $\omega_{\mathbb{2}}$ is infinite 'relative' to $\omega_{\mathbb{1}}$ and finite 'relative' to $\omega_{\mathbb{3}}$.

The nonstandard Turing hierarchy

In classical NSA, a number is either finite or infinite.

In 'stratified' or 'relative' NSA, there are 'levels' of infinity.
Consider $\mathbb{N}^{\mathbb{A}}$, an extension of ${}^*\mathbb{N}$:



where $\mathbb{A} = \{\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{3}, \ldots\}$

The infinite numbers are 'stratified' in $|\mathbb{A}|$ many levels of infinity.

Then $\omega_{\mathbb{2}}$ is infinite 'relative' to $\omega_{\mathbb{1}}$ and finite 'relative' to $\omega_{\mathbb{3}}$.

We write $\omega_{\mathbb{1}} \ll \omega_{\mathbb{2}} \ll \omega_{\mathbb{3}}$.

Transfer

$=$‘All levels $\mathbb{a} \in \mathbb{A}$ have the same properties as $\mathbb{N}$.’

Transfer

=‘All levels $\mathbb{a} \in \mathbb{A}$ have the same properties as $\mathbb{N}$.’

Define $\mathbb{N}^{\mathbb{a}} = \{n \in \mathbb{N}^{\mathbb{A}} | n \text{ is } \mathbb{a}\text{-finite}\}$.

Transfer

='All levels $\mathbb{a} \in \mathbb{A}$ have the same properties as $\mathbb{N}$.'

Define $\mathbb{N}^{\mathbb{a}} = \{n \in \mathbb{N}^{\mathbb{A}} | n \text{ is } \mathbb{a}\text{-finite}\}$.

Axiom schema ($\mathbf{\Pi_n}$)

For all $\varphi \in \Delta_0$ *and* $\mathbb{a} \in \mathbb{A}$

$$(\forall x_1 \in \mathbb{N})(\exists x_2 \in \mathbb{N}) \ldots (Qx_n \in \mathbb{N})\varphi(x_1, \ldots, x_n)$$

is equivalent to

Transfer

=‘All levels $\mathbb{a} \in \mathbb{A}$ have the same properties as $\mathbb{N}$.’

Define $\mathbb{N}^{\mathbb{a}} = \{n \in \mathbb{N}^{\mathbb{A}} | n \text{ is } \mathbb{a}\text{-finite}\}$.

Axiom schema (Π_n)

For all $\varphi \in \Delta_0$ *and* $\mathbb{a} \in \mathbb{A}$

$$(\forall x_1 \in \mathbb{N})(\exists x_2 \in \mathbb{N}) \dots (Qx_n \in \mathbb{N})\varphi(x_1, \dots, x_n)$$

is equivalent to

$$(\forall x_1 \in \mathbb{N}^{\mathbb{a}})(\exists x_2 \in \mathbb{N}^{\mathbb{a}}) \dots (Qx_n \in \mathbb{N}^{\mathbb{a}})\varphi(x_1, \dots, x_n)$$

The nonstandard Turing hierarchy

Theorem (Π_n-reduction)

Given $\mathbf{\Pi_n}$, for every $\varphi \in \Delta_0$,

$$(\forall x_1 \in \mathbb{N})(\exists x_2 \in \mathbb{N}) \ldots (Q x_n \in \mathbb{N}) \varphi(x_1, \ldots, x_n)$$

The nonstandard Turing hierarchy

Theorem (Π_n-reduction)

Given $\mathbf{\Pi_n}$, *for every* $\varphi \in \Delta_0$,

$$(\forall x_1 \in \mathbb{N})(\exists x_2 \in \mathbb{N}) \ldots (Qx_n \in \mathbb{N})\varphi(x_1, \ldots, x_n)$$

is equivalent to

$$(\forall x_1 \leq \omega_{\mathbb{1}})(\exists x_2 \leq \omega_{\mathbb{2}}) \ldots (Qx_n \leq \omega_{\mathbb{n}})\varphi(x_1, \ldots, x_n),$$

The nonstandard Turing hierarchy

Theorem (Π_n-reduction)

Given $\mathbf{\Pi_n}$, *for every* $\varphi \in \Delta_0$,

$$(\forall x_1 \in \mathbb{N})(\exists x_2 \in \mathbb{N}) \dots (Q x_n \in \mathbb{N}) \varphi(x_1, \dots, x_n)$$

is equivalent to

$$(\forall x_1 \leq \omega_{\mathbb{1}})(\exists x_2 \leq \omega_{\mathbb{2}}) \dots (Q x_n \leq \omega_{\mathbb{n}}) \varphi(x_1, \dots, x_n), \qquad (*)$$

with $\omega_{\mathbb{1}} \ll \omega_{\mathbb{2}} \ll \omega_{\mathbb{3}} \ll \dots \ll \omega_{\mathbb{n}}$.

The nonstandard Turing hierarchy

Theorem (Π_n-reduction)

Given $\mathbf{\Pi_n}$, *for every* $\varphi \in \Delta_0$,

$$(\forall x_1 \in \mathbb{N})(\exists x_2 \in \mathbb{N}) \dots (Qx_n \in \mathbb{N})\varphi(x_1, \dots, x_n)$$

is equivalent to

$$(\forall x_1 \leq \omega_{\mathbb{1}})(\exists x_2 \leq \omega_{\mathbb{2}}) \dots (Qx_n \leq \omega_{\mathbb{n}})\varphi(x_1, \dots, x_n), \qquad (*)$$

with $\omega_{\mathbb{1}} \ll \omega_{\mathbb{2}} \ll \omega_{\mathbb{3}} \ll \dots \ll \omega_{\mathbb{n}}$.

But $(*)$ is decidable!

The nonstandard Turing hierarchy

Theorem (Π_n-reduction)

Given $\mathbf{\Pi_n}$, *for every* $\varphi \in \Delta_0$,

$$(\forall x_1 \in \mathbb{N})(\exists x_2 \in \mathbb{N}) \dots (Q x_n \in \mathbb{N}) \varphi(x_1, \dots, x_n)$$

is equivalent to

$$(\forall x_1 \leq \omega_{\mathbb{1}})(\exists x_2 \leq \omega_{\mathbb{2}}) \dots (Q x_n \leq \omega_{\mathbb{n}}) \varphi(x_1, \dots, x_n), \qquad (*)$$

with $\omega_{\mathbb{1}} \ll \omega_{\mathbb{2}} \ll \omega_{\mathbb{3}} \ll \dots \ll \omega_{\mathbb{n}}$.

But $(*)$ is decidable!

Theorem

$A \leq_T \mathbf{0}^{(n)} \iff A \in \Delta_{n+1}$

The nonstandard Turing hierarchy

Theorem (Π_n-reduction)

Given $\mathbf{\Pi_n}$, *for every* $\varphi \in \Delta_0$,

$$(\forall x_1 \in \mathbb{N})(\exists x_2 \in \mathbb{N}) \ldots (Qx_n \in \mathbb{N})\varphi(x_1, \ldots, x_n)$$

is equivalent to

$$(\forall x_1 \leq \omega_{\mathbb{1}})(\exists x_2 \leq \omega_{\mathbb{2}}) \ldots (Qx_n \leq \omega_{\mathbb{n}})\varphi(x_1, \ldots, x_n), \qquad (*)$$

with $\omega_{\mathbb{1}} \ll \omega_{\mathbb{2}} \ll \omega_{\mathbb{3}} \ll \ldots \ll \omega_{\mathbb{n}}$.

But $(*)$ is decidable!

Theorem

$A \leq_T \mathbf{0}^{(n)} \iff A \in \Delta_{n+1} \iff A \leq_T \mathbf{\Pi_n}$.

Final Thoughts

Final Thoughts

The two eyes of exact science are mathematics and logic, the mathematical sect puts out the logical eye, the logical sect puts out the mathematical eye; each believing that it sees better with one eye than with two.

Augustus De Morgan

Final Thoughts

The two eyes of exact science are mathematics and logic, the mathematical sect puts out the logical eye, the logical sect puts out the mathematical eye; each believing that it sees better with one eye than with two.

Augustus De Morgan

...there are good reasons to believe that Nonstandard Analysis, in some version or other, will be the analysis of the future.

Kurt Gödel

Final Thoughts

The two eyes of exact science are mathematics and logic, the mathematical sect puts out the logical eye, the logical sect puts out the mathematical eye; each believing that it sees better with one eye than with two.

Augustus De Morgan

...there are good reasons to believe that Nonstandard Analysis, in some version or other, will be the analysis of the future.

Kurt Gödel

Thank you for your attention!

Final Thoughts

The two eyes of exact science are mathematics and logic, the mathematical sect puts out the logical eye, the logical sect puts out the mathematical eye; each believing that it sees better with one eye than with two.

Augustus De Morgan

...there are good reasons to believe that Nonstandard Analysis, in some version or other, will be the analysis of the future.

Kurt Gödel

Thank you for your attention!

Any questions?