Reuniting the Antipodes:

Bringing together Nonstandard and Constructive Analysis

Sam Sanders[1]

Madison, WI, April 2, 2012

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

Motivation

From the scope of CCA 2012: (http://cca-net.de/cca2012/)

Motivation

From the scope of CCA 2012: (http://cca-net.de/cca2012/)

The conference is concerned with the theory of computability and complexity over real-valued data.

Motivation

From the scope of CCA 2012: (http://cca-net.de/cca2012/)

The conference is concerned with the theory of computability and complexity over real-valued data. [BECAUSE]

Motivation

From the scope of CCA 2012: (http://cca-net.de/cca2012/)

The conference is concerned with the theory of computability and complexity over real-valued data. [BECAUSE] Most mathematical models in physics and engineering [...] are based on the real number concept.

Motivation

From the scope of CCA 2012: (http://cca-net.de/cca2012/)

The conference is concerned with the theory of computability and complexity over real-valued data. [BECAUSE] Most mathematical models in physics and engineering [...] are based on the real number concept.

The following is more true:

Most mathematical models in physics and engineering [...] are based on the real number concept, and an intuitive calculus with infinitesimals, i.e. informal Nonstandard Analysis.

Motivation

From the scope of CCA 2012: (http://cca-net.de/cca2012/)

The conference is concerned with the theory of computability and complexity over real-valued data. [BECAUSE] Most mathematical models in physics and engineering [...] are based on the real number concept.

The following is more true:

Most mathematical models in physics and engineering [...] are based on the real number concept, and an intuitive calculus with infinitesimals, i.e. informal Nonstandard Analysis.

We present a notion of constructive computability directly based on Nonstandard Analysis.

Constructive Analysis

Errett Bishop's Constructive Analysis is a redevelopment of Mathematics, consistent with CLASS, RUSS and INT.

Constructive Analysis

Errett Bishop's Constructive Analysis is a redevelopment of Mathematics, consistent with CLASS, RUSS and INT.

Definition (Logical connectives in BISH: BHK)

1. $P \vee Q$: we have an algorithm that outputs either P or Q, together with a proof of the chosen disjunct.

Constructive Analysis

Errett Bishop's Constructive Analysis is a redevelopment of Mathematics, consistent with CLASS, RUSS and INT.

Definition (Logical connectives in BISH: BHK)

1. $P \vee Q$: we have an algorithm that outputs either P or Q, together with a proof of the chosen disjunct.
2. $P \wedge Q$: we have both a proof of P and a proof of Q.

Constructive Analysis

Errett Bishop's Constructive Analysis is a redevelopment of Mathematics, consistent with CLASS, RUSS and INT.

Definition (Logical connectives in BISH: BHK)

1. $P \vee Q$: we have an algorithm that outputs either P or Q, together with a proof of the chosen disjunct.
2. $P \wedge Q$: we have both a proof of P and a proof of Q.
3. $P \to Q$: by means of an algorithm we can convert any proof of P into a proof of Q.

Constructive Analysis

Errett Bishop's Constructive Analysis is a redevelopment of Mathematics, consistent with CLASS, RUSS and INT.

Definition (Logical connectives in BISH: BHK)

1. $P \vee Q$: we have an algorithm that outputs either P or Q, together with a proof of the chosen disjunct.
2. $P \wedge Q$: we have both a proof of P and a proof of Q.
3. $P \to Q$: by means of an algorithm we can convert any proof of P into a proof of Q.
4. $\neg P \equiv P \to (0 = 1)$.

Constructive Analysis

Errett Bishop's Constructive Analysis is a redevelopment of Mathematics, consistent with CLASS, RUSS and INT.

Definition (Logical connectives in BISH: BHK)

1. $P \vee Q$: we have an algorithm that outputs either P or Q, together with a proof of the chosen disjunct.
2. $P \wedge Q$: we have both a proof of P and a proof of Q.
3. $P \to Q$: by means of an algorithm we can convert any proof of P into a proof of Q.
4. $\neg P \equiv P \to (0 = 1)$.
5. $(\exists x)P(x)$: an algorithm computes an object x_0 such that $P(x_0)$

Constructive Analysis

Errett Bishop's Constructive Analysis is a redevelopment of Mathematics, consistent with CLASS, RUSS and INT.

Definition (Logical connectives in BISH: BHK)

1. $P \vee Q$: we have an algorithm that outputs either P or Q, together with a proof of the chosen disjunct.
2. $P \wedge Q$: we have both a proof of P and a proof of Q.
3. $P \to Q$: by means of an algorithm we can convert any proof of P into a proof of Q.
4. $\neg P \equiv P \to (0 = 1)$.
5. $(\exists x)P(x)$: an algorithm computes an object x_0 such that $P(x_0)$
6. $(\forall x \in A)P(x)$: for all x, $x \in A \to P(x)$.

Constructive Analysis

Errett Bishop's Constructive Analysis is a redevelopment of Mathematics, consistent with CLASS, RUSS and INT.

Definition (Logical connectives in BISH: BHK)

1. $P \vee Q$: we have an algorithm that outputs either P or Q, together with a proof of the chosen disjunct.
2. $P \wedge Q$: we have both a proof of P and a proof of Q.
3. $P \rightarrow Q$: by means of an algorithm we can convert any proof of P into a proof of Q.
4. $\neg P \equiv P \rightarrow (0 = 1)$.
5. $(\exists x)P(x)$: an algorithm computes an object x_0 such that $P(x_0)$
6. $(\forall x \in A)P(x)$: for all x, $x \in A \rightarrow P(x)$.

From BISH to NSA

In BISH, proof and algorithm are central.

From BISH to NSA

In BISH, proof and algorithm are central.

We will define a system of Nonstandard Analyis $\mathbb{NSA}$ where transfer and Ω-invariant procedure play the same role.

From BISH to NSA

In BISH, proof and algorithm are central.

We will define a system of Nonstandard Analyis $\mathbb{NSA}$ where transfer and Ω-invariant procedure play the same role.

$\mathbb{NSA}$ is similar to $^*\mathsf{RCA}_0$, a nonstandard extension of RCA_0.

From BISH to NSA

In BISH, proof and algorithm are central.

We will define a system of Nonstandard Analyis $\mathbb{NSA}$ where transfer and Ω-invariant procedure play the same role.

$\mathbb{NSA}$ is similar to $^*\mathrm{RCA}_0$, a nonstandard extension of RCA_0.

Three important features:

From BISH to NSA

In BISH, proof and algorithm are central.

We will define a system of Nonstandard Analyis $\mathbb{NSA}$ where transfer and Ω-invariant procedure play the same role.

$\mathbb{NSA}$ is similar to $^*\mathsf{RCA}_0$, a nonstandard extension of RCA_0.

Three important features:

1. No transfer principle / elementary extension (except for Δ_0).

From BISH to NSA

In BISH, proof and algorithm are central.

We will define a system of Nonstandard Analyis $\mathbb{NSA}$ where transfer and Ω-invariant procedure play the same role.

$\mathbb{NSA}$ is similar to ${}^*\mathrm{RCA}_0$, a nonstandard extension of RCA_0.

Three important features:

1. No transfer principle / elementary extension (except for Δ_0).
2. No Δ_1-CA, but Ω-CA.

From BISH to NSA

In BISH, proof and algorithm are central.

We will define a system of Nonstandard Analyis $\mathbb{NSA}$ where transfer and Ω-invariant procedure play the same role.

$\mathbb{NSA}$ is similar to $^*\mathrm{RCA}_0$, a nonstandard extension of RCA_0.

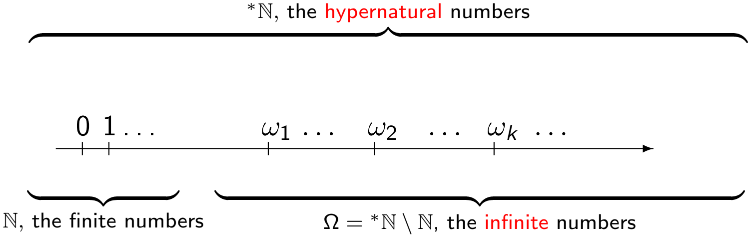
Three important features:

1. No transfer principle / elementary extension (except for Δ_0).
2. No Δ_1-CA, but Ω-CA.
3. Levels of infinity (Stratified NSA).

Feature 3: Stratified Nonstandard Analysis

Feature 3: Stratified Nonstandard Analysis

The usual picture of ${}^*\mathbb{N}$:

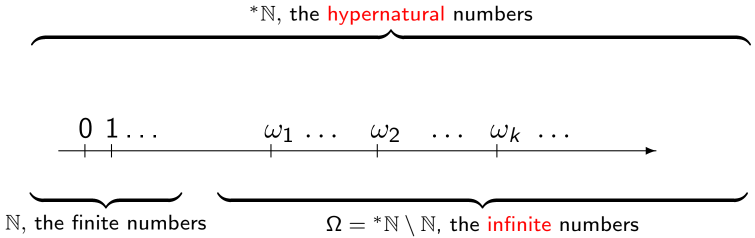


Feature 3: Stratified Nonstandard Analysis

The usual picture of ${}^*\mathbb{N}$:

In $\mathbb{NSA}$, ${}^*\mathbb{N}$ has extra structure:

countably many levels of infinity $\mathbb{N} \subset \mathbb{N}_1 \subset \ldots \mathbb{N}_k \subset \mathbb{N}_{k+1} \subset \cdots \subset {}^*\mathbb{N}$

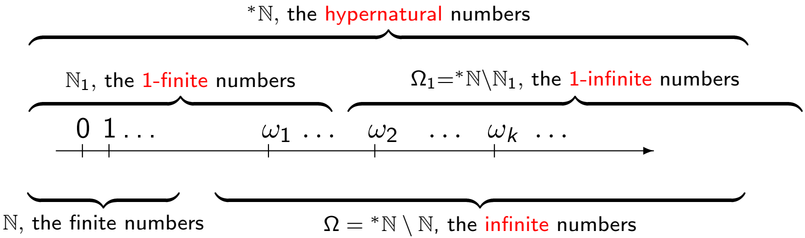


Feature 3: Stratified Nonstandard Analysis

The usual picture of ${}^*\mathbb{N}$:

In $\mathbb{NSA}$, ${}^*\mathbb{N}$ has extra structure:

countably many levels of infinity $\mathbb{N} \subset \mathbb{N}_1 \subset \dots \mathbb{N}_k \subset \mathbb{N}_{k+1} \subset \dots \subset {}^*\mathbb{N}$

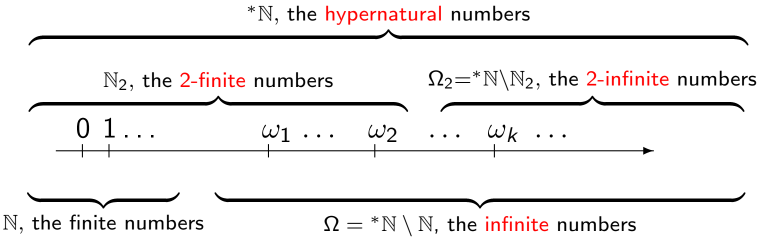


Feature 3: Stratified Nonstandard Analysis

The usual picture of ${}^*\mathbb{N}$:

In $\mathbb{NSA}$, ${}^*\mathbb{N}$ has extra structure:

countably many levels of infinity $\mathbb{N} \subset \mathbb{N}_1 \subset \dots \mathbb{N}_k \subset \mathbb{N}_{k+1} \subset \cdots \subset {}^*\mathbb{N}$

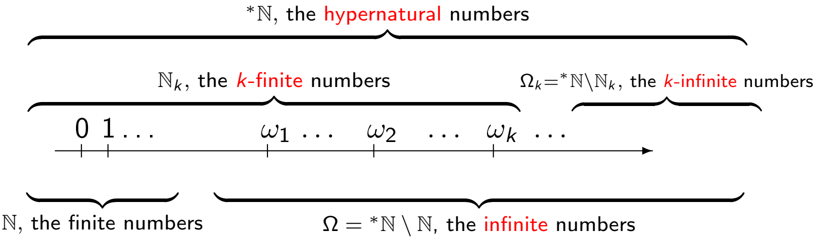


Feature 3: Stratified Nonstandard Analysis

The usual picture of ${}^*\mathbb{N}$:

In $\mathbb{NSA}$, ${}^*\mathbb{N}$ has extra structure:

countably many levels of infinity $\mathbb{N} \subset \mathbb{N}_1 \subset \dots \mathbb{N}_k \subset \mathbb{N}_{k+1} \subset \cdots \subset {}^*\mathbb{N}$



Feature 2: Ω-invariance

Feature 2: Ω-invariance

Ω-invariance $\approx$ algorithm $\approx$ finite procedure $\approx$ explicit computation.

Feature 2: Ω-invariance

Ω-invariance ≈ algorithm ≈ finite procedure ≈ explicit computation.

Definition (Ω-invariance)

A set $A \subset \mathbb{N}$ is Ω-invariant

Feature 2: Ω-invariance

Ω-invariance $\approx$ algorithm $\approx$ finite procedure $\approx$ explicit computation.

Definition (Ω-invariance)

A set $A \subset \mathbb{N}$ is Ω-invariant if there is a quantifier-free formula ψ such that for all $\omega \in \Omega$,

Feature 2: Ω-invariance

Ω-invariance $\approx$ algorithm $\approx$ finite procedure $\approx$ explicit computation.

Definition (Ω-invariance)

A set $A \subset \mathbb{N}$ is Ω-invariant if there is a quantifier-free formula ψ such that for all $\omega \in \Omega$,

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

Feature 2: Ω-invariance

Ω-invariance $\approx$ algorithm $\approx$ finite procedure $\approx$ explicit computation.

Definition (Ω-invariance)

A set $A \subset \mathbb{N}$ is Ω-invariant if there is a quantifier-free formula ψ such that for all $\omega \in \Omega$,

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

Note that A depends on $\omega \in \Omega$, but not on the choice of $\omega \in \Omega$.

Feature 2: Ω-invariance

Theorem (Finiteness)

For every Ω-invariant $A \subset \mathbb{N}$ and $k \in \mathbb{N}$, there is $M \in \mathbb{N}$, such that

Feature 2: Ω-invariance

Theorem (Finiteness)

For every Ω-invariant $A \subset \mathbb{N}$ and $k \in \mathbb{N}$, there is $M \in \mathbb{N}$, such that

$$k \in A \iff \psi(k, \omega) \iff \psi(k, M).$$

Feature 2: Ω-invariance

Theorem (Finiteness)

For every Ω-invariant $A \subset \mathbb{N}$ and $k \in \mathbb{N}$, there is $M \in \mathbb{N}$, such that

$$k \in A \iff \psi(k, \omega) \iff \psi(k, M).$$

Thus, to verify whether $k \in A$, we only need to perform finitely many operations (i.e. determine if $\psi(k, M)$).

Feature 2: Ω-invariance

Theorem (Finiteness)

For every Ω-invariant $A \subset \mathbb{N}$ and $k \in \mathbb{N}$, there is $M \in \mathbb{N}$, such that

$$k \in A \Longleftrightarrow \psi(k, \omega) \Longleftrightarrow \psi(k, M).$$

Thus, to verify whether $k \in A$, we only need to perform finitely many operations (i.e. determine if $\psi(k, M)$).

$\mathbb{NSA}$ has Ω-CA instead of Δ_1-CA.

Principle (Ω-CA)

All Ω-invariant sets exist.

Lost in translation

Lost in translation

BISH (based on IL)

Lost in translation

BISH (based on IL)	NSA (based on CL)

Lost in translation

BISH (based on IL)

Central: algorithm and proof

NSA (based on CL)

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	
$A \vee B$: an algo yields a proof of A or of B	

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	

Lost in translation

BISH (based on IL)

Central: algorithm and proof

$A \vee B$:
an algo yields a proof of A or of B

NSA (based on CL)

Central: Ω-invariance and transfer ($\mathbb{T}$)

$A \mathbb{V} B$:
$[A \vee B] \wedge [A \rightarrow A \in \mathbb{T}] \wedge [B \rightarrow B \in \mathbb{T}]$

Lost in translation

BISH (based on IL)

Central: algorithm and proof

$A \vee B$:
an algo yields a proof of A or of B

$A \to B$: an algo converts a proof of A to a proof of B

NSA (based on CL)

Central: Ω-invariance and transfer ($\mathbb{T}$)

$A \mathbb{V} B$:
$[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \to B \wedge [B \in \mathbb{T}]$

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \to B \wedge [B \in \mathbb{T}]$
	'$A \in \mathbb{T}$' means 'A satisfies Transfer'.

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \to B \wedge [B \in \mathbb{T}]$
	'$A \in \mathbb{T}$' means 'A satisfies Transfer'.

E.g. $(\forall n \in \mathbb{N})\varphi(n) \in \mathbb{T}$ is $[(\forall n \in \mathbb{N})\varphi(n) \to (\forall n \in {}^*\mathbb{N})\varphi(n)]$

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \to B \wedge [B \in \mathbb{T}]$
	'$A \in \mathbb{T}$' means 'A satisfies Transfer'.

E.g. $(\forall n \in \mathbb{N})\varphi(n) \in \mathbb{T}$ is $[(\forall n \in \mathbb{N})\varphi(n) \to (\forall n \in {}^*\mathbb{N})\varphi(n)]$

E.g. $(\exists n \in {}^*\mathbb{N})\varphi(n) \in \mathbb{T}$ is $[(\exists n \in {}^*\mathbb{N})\varphi(n) \to (\exists n \in \mathbb{N}_1)\varphi(n)]$

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \rightarrow A \in \mathbb{T}] \wedge [B \rightarrow B \in \mathbb{T}]$
$A \rightarrow B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \rightarrow B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \rightarrow (0 = 1)$	

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \rightarrow A \in \mathbb{T}] \wedge [B \rightarrow B \in \mathbb{T}]$
$A \rightarrow B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \rightarrow B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \rightarrow (0 = 1)$	$\sim A$: $A \Rrightarrow (0 = 1)$

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \mathbin{\mathbb{V}} B$: $[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \to B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \to (0 = 1)$	$\sim A$: $A \Rrightarrow (0 = 1)$

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer $(\mathbb{T})$
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$
$\approx$ "an algo decides if A or if B"	
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \to B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \to (0 = 1)$	$\sim A$: $A \Rrightarrow (0 = 1)$

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$
$\approx$ "an algo decides if A or if B"	$\approx$ "an Ω-inv.proc. decides if A or if B"
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \to B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \to (0 = 1)$	$\sim A$: $A \Rrightarrow (0 = 1)$

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer $(\mathbb{T})$
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$
$\approx$ "an algo decides if A or if B"	$\approx$ "an Ω-inv.proc. decides if A or if B"
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \to B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \to (0 = 1)$	$\sim A$: $A \Rrightarrow (0 = 1)$
$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$	

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \rightarrow A \in \mathbb{T}] \wedge [B \rightarrow B \in \mathbb{T}]$
$\approx$ "an algo decides if A or if B"	$\approx$ "an Ω-inv.proc. decides if A or if B"
$A \rightarrow B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \rightarrow B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \rightarrow (0 = 1)$	$\sim A$: $A \Rrightarrow (0 = 1)$
$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$	$(\exists x)A(x)$: an Ω-inv. proc. computes x_0 such that $A(x_0)$

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$
$\approx$ "an algo decides if A or if B"	$\approx$ "an Ω-inv.proc. decides if A or if B"
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \to B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \to (0 = 1)$	$\sim A$: $A \Rrightarrow (0 = 1)$
$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$	$(\exists x)A(x)$: an Ω-inv. proc. computes x_0 such that $A(x_0)$
	$\sim[(\forall n \in \mathbb{N})A(n)]$

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer $(\mathbb{T})$
$A \vee B$: an algo yields a proof of A or of B	$A \mathbin{\mathbb{V}} B$: $[A \vee B] \wedge [A \rightarrow A \in \mathbb{T}] \wedge [B \rightarrow B \in \mathbb{T}]$
$\approx$ "an algo decides if A or if B"	$\approx$ "an Ω-inv.proc. decides if A or if B"
$A \rightarrow B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \rightarrow B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \rightarrow (0 = 1)$	$\sim A$: $A \Rrightarrow (0 = 1)$
$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$	$(\exists x)A(x)$: an Ω-inv. proc. computes x_0 such that $A(x_0)$
	$\sim[(\forall n \in \mathbb{N})A(n)] \equiv (\exists n \in \mathbb{N}_1)\neg A(n)$
	WEAKER than $(\exists n \in \mathbb{N})\neg A(n)$.

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \,\mathbb{V}\, B$: $[A \vee B] \wedge [A \rightarrow A \in \mathbb{T}] \wedge [B \rightarrow B \in \mathbb{T}]$
$\approx$ "an algo decides if A or if B"	$\approx$ "an Ω-inv.proc. decides if A or if B"
$A \rightarrow B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \rightarrow B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \rightarrow (0 = 1)$	$\sim A$: $A \Rrightarrow (0 = 1)$
$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$	$(\exists x)A(x)$: an Ω-inv. proc. computes x_0 such that $A(x_0)$
$\neg[(\forall n \in \mathbb{N})A(n)]$ is WEAKER than $(\exists n \in \mathbb{N})\neg A(n)$.	$\sim[(\forall n \in \mathbb{N})A(n)] \equiv (\exists n \in \mathbb{N}_1)\neg A(n)$ WEAKER than $(\exists n \in \mathbb{N})\neg A(n)$.

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \mathbb{V} B$: $[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$
$\approx$ "an algo decides if A or if B"	$\approx$ "an Ω-inv.proc. decides if A or if B"
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \to B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \to (0 = 1)$	$\sim A$: $A \Rrightarrow (0 = 1)$
$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$	$(\exists x)A(x)$: an Ω-inv. proc. computes x_0 such that $A(x_0)$

WHY is this a good/faithful/reasonable/... translation?

Lost in translation

BISH (based on IL)	NSA (based on CL)
Central: algorithm and proof	Central: Ω-invariance and transfer ($\mathbb{T}$)
$A \vee B$: an algo yields a proof of A or of B	$A \,\mathbb{V}\, B$: $[A \vee B] \wedge [A \to A \in \mathbb{T}] \wedge [B \to B \in \mathbb{T}]$
$\approx$ "an algo decides if A or if B"	$\approx$ "an Ω-inv.proc. decides if A or if B"
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rrightarrow B$: $A \wedge [A \in \mathbb{T}] \to B \wedge [B \in \mathbb{T}]$
$\neg A$: $A \to (0 = 1)$	$\sim A$: $A \Rrightarrow (0 = 1)$
$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$	$(\exists x)A(x)$: an Ω-inv. proc. computes x_0 such that $A(x_0)$

WHY is this a good/faithful/reasonable/... translation?

BECAUSE the non-algorithmic/non-constructive principles behave the same!

Constructive Reverse Mathematics

Constructive Reverse Mathematics

BISH (based on IL)

non-constructive/non-algorithmic

NSA (based on CL)

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	
$\updownarrow$	

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	
$\updownarrow$	
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	
$\updownarrow$	

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	
$\updownarrow$	
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	
$\updownarrow$	
MCT: monotone convergence thm	
$\updownarrow$	

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	
$\updownarrow$	
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	
$\updownarrow$	
MCT: monotone convergence thm	
$\updownarrow$	
CIT: Cantor intersection thm	

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant

LPO: For $P \in \Sigma_1$, $P \vee \neg P$

$\updownarrow$

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

$\updownarrow$

MCT: monotone convergence thm

$\updownarrow$

CIT: Cantor intersection thm

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} \sim P$
$\updownarrow$	$\updownarrow$
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	
$\updownarrow$	
MCT: monotone convergence thm	
$\updownarrow$	
CIT: Cantor intersection thm	

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} \sim P$
$\updownarrow$	$\updownarrow$
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbb{V} \sim(x < 0))$
$\updownarrow$	$\updownarrow$
MCT: monotone convergence thm	
$\updownarrow$	
CIT: Cantor intersection thm	

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} \sim P$
$\updownarrow$	$\updownarrow$
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbb{V} \sim(x < 0))$
$\updownarrow$	$\updownarrow$
MCT: monotone convergence thm	$\mathbb{MCT}$: monotone convergence thm
$\updownarrow$	$\updownarrow$
CIT: Cantor intersection thm	

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} {\sim} P$
$\updownarrow$	$\updownarrow$
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbb{V} {\sim}(x < 0))$
$\updownarrow$	$\updownarrow$
MCT: monotone convergence thm	$\mathbb{MCT}$: monotone convergence thm
$\updownarrow$	$\updownarrow$
CIT: Cantor intersection thm	$\mathbb{CIT}$: Cantor intersection thm

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} \sim P$
$\updownarrow$	$\updownarrow$
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbb{V} \sim(x < 0))$
$\updownarrow$	$\updownarrow$
MCT: monotone convergence thm	$\mathbb{MCT}$: monotone convergence thm
$\updownarrow$ (limit computed by algo)	$\updownarrow$
CIT: Cantor intersection thm	$\mathbb{CIT}$: Cantor intersection thm

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} \sim P$
$\updownarrow$	$\updownarrow$
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbb{V} \sim (x < 0))$
$\updownarrow$	$\updownarrow$
MCT: monotone convergence thm	$\mathbb{MCT}$: monotone convergence thm
$\updownarrow$ (limit computed by algo)	$\updownarrow$ (limit computed by Ω-inv. proc.)
CIT: Cantor intersection thm	$\mathbb{CIT}$: Cantor intersection thm

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} {\sim} P$
$\updownarrow$	$\updownarrow$
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbb{V} {\sim}(x < 0))$
$\updownarrow$	$\updownarrow$
MCT: monotone convergence thm	$\mathbb{MCT}$: monotone convergence thm
$\updownarrow$ (limit computed by algo)	$\updownarrow$ (limit computed by Ω-inv. proc.)
CIT: Cantor intersection thm	$\mathbb{CIT}$: Cantor intersection thm
(point in intersection computed by algo)	

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbin{\mathbb{V}} \sim P$
$\updownarrow$	$\updownarrow$
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbin{\mathbb{V}} \sim(x < 0))$
$\updownarrow$	$\updownarrow$
MCT: monotone convergence thm	$\mathbb{MCT}$: monotone convergence thm
$\updownarrow$ (limit computed by algo)	$\updownarrow$ (limit computed by Ω-inv. proc.)
CIT: Cantor intersection thm	$\mathbb{CIT}$: Cantor intersection thm
(point in intersection computed by algo)	(point in intersection computed by Ω-inv. proc.)

Constructive Reverse Mathematics

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbin{\mathbb{V}} \sim P$
$\updownarrow$	$\updownarrow$
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbin{\mathbb{V}} \sim(x < 0))$
$\updownarrow$	$\updownarrow$
MCT: monotone convergence thm	$\mathbb{MCT}$: monotone convergence thm
$\updownarrow$ (limit computed by algo)	$\updownarrow$ (limit computed by Ω-inv. proc.)
CIT: Cantor intersection thm	$\mathbb{CIT}$: Cantor intersection thm
	$\updownarrow$
	Universal Transfer
	$(\forall n \in \mathbb{N})\varphi(n) \to (\forall n \in {}^*\mathbb{N})\varphi(n)$

Constructive Reverse Mathematics II

Constructive Reverse Mathematics II

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	

Constructive Reverse Mathematics II

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	
LLPO	
For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$	
$\updownarrow$	

Constructive Reverse Mathematics II

BISH (based on IL)

non-constructive/non-algorithmic

LLPO

For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$

$\updownarrow$

LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$

$\updownarrow$

NSA (based on CL)

Constructive Reverse Mathematics II

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	

LLPO

For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$

$\updownarrow$

LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$

$\updownarrow$

NIL

$(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$

$\updownarrow$

Constructive Reverse Mathematics II

BISH (based on IL)

non-constructive/non-algorithmic

LLPO

For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$

$\updownarrow$

LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$

$\updownarrow$

NIL

$(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$

$\updownarrow$

IVT: Intermediate value theorem

NSA (based on CL)

Constructive Reverse Mathematics II

BISH (based on IL)

non-constructive/non-algorithmic

LLPO

For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \to \neg P \vee \neg Q$

$\updownarrow$

LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$

$\updownarrow$

NIL

$(\forall x, y \in \mathbb{R})(xy = 0 \to x = 0 \vee y = 0)$

$\updownarrow$

IVT: Intermediate value theorem

NSA (based on CL)

non-Ω-invariant

Constructive Reverse Mathematics II

BISH (based on IL)

non-constructive/non-algorithmic

LLPO

For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$

$\updownarrow$

LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$

$\updownarrow$

NIL

$(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$

$\updownarrow$

IVT: Intermediate value theorem

NSA (based on CL)

non-Ω-invariant

$\mathbb{LLPO}$

For $P, Q \in \Sigma_1$, $\sim(P \wedge Q) \Rrightarrow \sim P \mathbb{V} \sim Q$

$\updownarrow$

Constructive Reverse Mathematics II

BISH (based on IL)

non-constructive/non-algorithmic

LLPO

For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$

$\updownarrow$

LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$

$\updownarrow$

NIL

$(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$

$\updownarrow$

IVT: Intermediate value theorem

NSA (based on CL)

non-Ω-invariant

$\mathbb{LLPO}$

For $P, Q \in \Sigma_1$, $\sim(P \wedge Q) \Rrightarrow \sim P \,\mathbb{V} \sim Q$

$\updownarrow$

$\mathbb{LLPR}$: $(\forall x \in \mathbb{R})(x \geq 0 \,\mathbb{V}\, x \leq 0)$

$\updownarrow$

Constructive Reverse Mathematics II

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LLPO	$\mathbb{LLPO}$
For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$	For $P, Q \in \Sigma_1$, $\sim(P \wedge Q) \Rrightarrow \sim P \mathbb{V} \sim Q$
$\updownarrow$	$\updownarrow$
LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$	$\mathbb{LLPR}$: $(\forall x \in \mathbb{R})(x \geq 0 \mathbb{V} x \leq 0)$
$\updownarrow$	$\updownarrow$
NIL	$\mathbb{NIL}$
$(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$	$(\forall x, y \in \mathbb{R})(xy = 0 \Rrightarrow x = 0 \mathbb{V} y = 0)$
$\updownarrow$	$\updownarrow$
IVT: Intermediate value theorem	

Constructive Reverse Mathematics II

BISH (based on IL)
non-constructive/non-algorithmic

LLPO

For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$

$\updownarrow$

LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$

$\updownarrow$

NIL

$(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$

$\updownarrow$

IVT: Intermediate value theorem

NSA (based on CL)
non-Ω-invariant

$\mathbb{LLPO}$

For $P, Q \in \Sigma_1$, $\sim(P \wedge Q) \Rrightarrow \sim P \,\mathbb{V}\, \sim Q$

$\updownarrow$

$\mathbb{LLPR}$: $(\forall x \in \mathbb{R})(x \geq 0 \,\mathbb{V}\, x \leq 0)$

$\updownarrow$

$\mathbb{NIL}$

$(\forall x, y \in \mathbb{R})(xy = 0 \Rrightarrow x = 0 \,\mathbb{V}\, y = 0)$

$\updownarrow$

$\mathbb{IVT}$: Intermediate value theorem

Constructive Reverse Mathematics II

BISH (based on IL)

non-constructive/non-algorithmic

LLPO

For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$

$\updownarrow$

LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$

$\updownarrow$

NIL

$(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$

$\updownarrow$

IVT: Intermediate value theorem

(int. value computed by algo)

NSA (based on CL)

non-Ω-invariant

$\mathbb{LLPO}$

For $P, Q \in \Sigma_1$, $\sim(P \wedge Q) \Rrightarrow \sim P \mathbb{V} \sim Q$

$\updownarrow$

$\mathbb{LLPR}$: $(\forall x \in \mathbb{R})(x \geq 0 \mathbb{V} x \leq 0)$

$\updownarrow$

$\mathbb{NIL}$

$(\forall x, y \in \mathbb{R})(xy = 0 \Rrightarrow x = 0 \mathbb{V} y = 0)$

$\updownarrow$

$\mathbb{IVT}$: Intermediate value theorem

Constructive Reverse Mathematics II

BISH (based on IL)

non-constructive/non-algorithmic

LLPO

For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$

$\updownarrow$

LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$

$\updownarrow$

NIL

$(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$

$\updownarrow$

IVT: Intermediate value theorem

(int. value computed by algo)

NSA (based on CL)

non-Ω-invariant

$\mathbb{LLPO}$

For $P, Q \in \Sigma_1$, $\sim(P \wedge Q) \Rrightarrow \sim P \,\mathbb{V} \sim Q$

$\updownarrow$

$\mathbb{LLPR}$: $(\forall x \in \mathbb{R})(x \geq 0 \,\mathbb{V}\, x \leq 0)$

$\updownarrow$

$\mathbb{NIL}$

$(\forall x, y \in \mathbb{R})(xy = 0 \Rrightarrow x = 0 \,\mathbb{V}\, y = 0)$

$\updownarrow$

$\mathbb{IVT}$: Intermediate value theorem

(int. value computed by Ω-inv. proc.)

Constructive Reverse Mathematics II

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LLPO	$\mathbb{LLPO}$
For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$	For $P, Q \in \Sigma_1$, $\sim(P \wedge Q) \Rrightarrow \sim P \mathbb{V} \sim Q$
$\updownarrow$	$\updownarrow$
LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$	$\mathbb{LLPR}$: $(\forall x \in \mathbb{R})(x \geq 0 \mathbb{V} x \leq 0)$
$\updownarrow$	$\updownarrow$
NIL	$\mathbb{NIL}$
$(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$	$(\forall x, y \in \mathbb{R})(xy = 0 \Rrightarrow x = 0 \mathbb{V} y = 0)$
$\updownarrow$	$\updownarrow$
IVT: Intermediate value theorem	$\mathbb{IVT}$: Intermediate value theorem
(int. value computed by algo)	(int. value computed by Ω-inv. proc.)
Axioms of $\mathbb{R}$: $\neg(x > 0 \wedge x < 0)$	

Constructive Reverse Mathematics II

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
LLPO	$\mathbb{LLPO}$
For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$	For $P, Q \in \Sigma_1$, $\sim(P \wedge Q) \Rrightarrow \sim P \mathbb{V} \sim Q$
$\updownarrow$	$\updownarrow$
LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$	$\mathbb{LLPR}$: $(\forall x \in \mathbb{R})(x \geq 0 \mathbb{V} x \leq 0)$
$\updownarrow$	$\updownarrow$
NIL	$\mathbb{NIL}$
$(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$	$(\forall x, y \in \mathbb{R})(xy = 0 \Rrightarrow x = 0 \mathbb{V} y = 0)$
$\updownarrow$	$\updownarrow$
IVT: Intermediate value theorem	$\mathbb{IVT}$: Intermediate value theorem
(int. value computed by algo)	(int. value computed by Ω-inv. proc.)
Axioms of $\mathbb{R}$: $\neg(x > 0 \wedge x < 0)$	Axioms of $\mathbb{R}$: $\sim(x > 0 \wedge x < 0)$

Constructive Reverse Mathematics III

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	
MP: For $P \in \Sigma_1$, $\neg\neg P \to P$	
$\updownarrow$	

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	

MP: For $P \in \Sigma_1$, $\neg\neg P \to P$

$\updownarrow$

MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \to x > 0)$

$\updownarrow$

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic MP: For $P \in \Sigma_1$, $\neg\neg P \to P$ $\updownarrow$ MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \to x > 0)$ $\updownarrow$ EXT: the extensionality theorem	

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
MP: For $P \in \Sigma_1$, $\neg\neg P \rightarrow P$	
$\updownarrow$	
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \rightarrow x > 0)$	
$\updownarrow$	
EXT: the extensionality theorem	

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
MP: For $P \in \Sigma_1$, $\neg\neg P \to P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rrightarrow P$
$\updownarrow$	$\updownarrow$
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \to x > 0)$	
$\updownarrow$	
EXT: the extensionality theorem	

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
MP: For $P \in \Sigma_1$, $\neg\neg P \to P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rrightarrow P$
$\updownarrow$	$\updownarrow$
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \to x > 0)$	$\mathbb{MPR}$: $(\forall x \in \mathbb{R})(\sim\sim(x > 0) \Rrightarrow x > 0)$
$\updownarrow$	$\updownarrow$
EXT: the extensionality theorem	

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
MP: For $P \in \Sigma_1$, $\neg\neg P \to P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rrightarrow P$
$\updownarrow$	$\updownarrow$
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \to x > 0)$	$\mathbb{MPR}$: $(\forall x \in \mathbb{R})(\sim\sim(x > 0) \Rrightarrow x > 0)$
$\updownarrow$	$\updownarrow$
EXT: the extensionality theorem	$\mathbb{EXT}$: the extensionality theorem

Constructive Reverse Mathematics III

BISH (based on IL) non-constructive/non-algorithmic	NSA (based on CL) non-Ω-invariant
MP: For $P \in \Sigma_1$, $\neg\neg P \rightarrow P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rrightarrow P$
$\updownarrow$	$\updownarrow$
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \rightarrow x > 0)$	$\mathbb{MPR}$: $(\forall x \in \mathbb{R})(\sim\sim(x > 0) \Rrightarrow x > 0)$
$\updownarrow$	$\updownarrow$
EXT: the extensionality theorem	$\mathbb{EXT}$: the extensionality theorem
WLPO: For $P \in \Sigma_1$, $\neg\neg P \vee \neg P$	
$\updownarrow$	

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
MP: For $P \in \Sigma_1$, $\neg\neg P \rightarrow P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rrightarrow P$
$\updownarrow$	$\updownarrow$
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \rightarrow x > 0)$	$\mathbb{MPR}$: $(\forall x \in \mathbb{R})(\sim\sim(x > 0) \Rrightarrow x > 0)$
$\updownarrow$	$\updownarrow$
EXT: the extensionality theorem	$\mathbb{EXT}$: the extensionality theorem
WLPO: For $P \in \Sigma_1$, $\neg\neg P \vee \neg P$	
$\updownarrow$	
WLPR: $(\forall x \in \mathbb{R})\big[\neg\neg(x > 0) \vee \neg(x > 0)\big]$	
$\updownarrow$	

Constructive Reverse Mathematics III

BISH (based on IL)

non-constructive/non-algorithmic

MP: For $P \in \Sigma_1$, $\neg\neg P \rightarrow P$

$\updownarrow$

MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \rightarrow x > 0)$

$\updownarrow$

EXT: the extensionality theorem

WLPO: For $P \in \Sigma_1$, $\neg\neg P \vee \neg P$

$\updownarrow$

WLPR: $(\forall x \in \mathbb{R})\big[\neg\neg(x > 0) \vee \neg(x > 0)\big]$

$\updownarrow$

DISC:

A discontinuous $2^{\mathbb{N}} \rightarrow \mathbb{N}$-function exists.

NSA (based on CL)

non-Ω-invariant

$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rrightarrow P$

$\updownarrow$

$\mathbb{MPR}$: $(\forall x \in \mathbb{R})(\sim\sim(x > 0) \Rrightarrow x > 0)$

$\updownarrow$

$\mathbb{EXT}$: the extensionality theorem

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
MP: For $P \in \Sigma_1$, $\neg\neg P \to P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rrightarrow P$
$\updownarrow$	$\updownarrow$
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \to x > 0)$	$\mathbb{MPR}$: $(\forall x \in \mathbb{R})(\sim\sim(x > 0) \Rrightarrow x > 0)$
$\updownarrow$	$\updownarrow$
EXT: the extensionality theorem	$\mathbb{EXT}$: the extensionality theorem
WLPO: For $P \in \Sigma_1$, $\neg\neg P \vee \neg P$	$\mathbb{WLPO}$: For $P \in \Sigma_1$, $\sim\sim P \,\mathbb{V} \sim P$
$\updownarrow$	$\updownarrow$
WLPR: $(\forall x \in \mathbb{R})\big[\neg\neg(x > 0) \vee \neg(x > 0)\big]$	
$\updownarrow$	
DISC: A discontinuous $2^{\mathbb{N}} \to \mathbb{N}$-function exists.	

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
MP: For $P \in \Sigma_1$, $\neg\neg P \rightarrow P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rrightarrow P$
$\updownarrow$	$\updownarrow$
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \rightarrow x > 0)$	$\mathbb{MPR}$: $(\forall x \in \mathbb{R})(\sim\sim(x > 0) \Rrightarrow x > 0)$
$\updownarrow$	$\updownarrow$
EXT: the extensionality theorem	$\mathbb{EXT}$: the extensionality theorem
WLPO: For $P \in \Sigma_1$, $\neg\neg P \vee \neg P$	$\mathbb{WLPO}$: For $P \in \Sigma_1$, $\sim\sim P \mathbb{V} \sim P$
$\updownarrow$	$\updownarrow$
WLPR: $(\forall x \in \mathbb{R})\big[\neg\neg(x > 0) \vee \neg(x > 0)\big]$	$\mathbb{WLPR}$: $(\forall x \in \mathbb{R})\big[\sim\sim(x > 0) \mathbb{V} \sim(x > 0)\big]$
$\updownarrow$	$\updownarrow$
DISC: A discontinuous $2^{\mathbb{N}} \rightarrow \mathbb{N}$-function exists.	

Constructive Reverse Mathematics III

BISH (based on IL)	NSA (based on CL)
non-constructive/non-algorithmic	non-Ω-invariant
MP: For $P \in \Sigma_1$, $\neg\neg P \to P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rrightarrow P$
$\updownarrow$	$\updownarrow$
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \to x > 0)$	$\mathbb{MPR}$: $(\forall x \in \mathbb{R})(\sim\sim(x > 0) \Rrightarrow x > 0)$
$\updownarrow$	$\updownarrow$
EXT: the extensionality theorem	$\mathbb{EXT}$: the extensionality theorem
WLPO: For $P \in \Sigma_1$, $\neg\neg P \vee \neg P$	$\mathbb{WLPO}$: For $P \in \Sigma_1$, $\sim\sim P \mathbb{V} \sim P$
$\updownarrow$	$\updownarrow$
WLPR: $(\forall x \in \mathbb{R})[\neg\neg(x > 0) \vee \neg(x > 0)]$	$\mathbb{WLPR}$: $(\forall x \in \mathbb{R})[\sim\sim(x > 0) \mathbb{V} \sim(x > 0)]$
$\updownarrow$	$\updownarrow$
DISC: A discontinuous $2^{\mathbb{N}} \to \mathbb{N}$-function exists.	$\mathbb{DISC}$: A discontinuous $2^{^*\mathbb{N}} \to {}^*\mathbb{N}$-function exists.

Constructive Reverse Mathematics

Constructive Reverse Mathematics

Same for WMP, FAN_Δ, BD-N, and $\mathsf{MP}^\vee$.

Constructive Reverse Mathematics

Same for WMP, FAN_Δ, BD-N, and $\text{MP}^\vee$.
Same for ‘mixed’ theorems:

Constructive Reverse Mathematics

Same for WMP, FAN_Δ, BD-N, and $\text{MP}^\vee$.
Same for 'mixed' theorems:

BISH (based on IL)	NSA (based on CL)

Constructive Reverse Mathematics

Same for WMP, FAN_Δ, BD-N, and $\mathrm{MP}^\vee$.
Same for 'mixed' theorems:

BISH (based on IL)	NSA (based on CL)
LPO $\leftrightarrow$ MP+WLPO	

Constructive Reverse Mathematics

Same for WMP, FAN_{Δ}, BD-N, and $MP^{\vee}$.
Same for 'mixed' theorems:

BISH (based on IL)	NSA (based on CL)
LPO $\leftrightarrow$ MP+WLPO MP $\leftrightarrow$ WMP + $MP^{\vee}$	

Constructive Reverse Mathematics

Same for WMP, FAN_Δ, BD-N, and $\text{MP}^\vee$.
Same for 'mixed' theorems:

BISH (based on IL)	NSA (based on CL)
LPO $\leftrightarrow$ MP+WLPO MP $\leftrightarrow$ WMP + $\text{MP}^\vee$ WLPO $\rightarrow$ LLPO	

Constructive Reverse Mathematics

Same for WMP, FAN_Δ, BD-N, and $\mathrm{MP}^\vee$.
Same for 'mixed' theorems:

BISH (based on IL)	NSA (based on CL)
LPO $\leftrightarrow$ MP+WLPO MP $\leftrightarrow$ WMP + $\mathrm{MP}^\vee$ WLPO $\rightarrow$ LLPO LLPO $\rightarrow$ $\mathrm{MP}^\vee$	

Constructive Reverse Mathematics

Same for WMP, FAN_Δ, BD-N, and $\mathrm{MP}^\vee$.
Same for 'mixed' theorems:

BISH (based on IL)	NSA (based on CL)
LPO $\leftrightarrow$ MP+WLPO MP $\leftrightarrow$ WMP + $\mathrm{MP}^\vee$ WLPO $\rightarrow$ LLPO LLPO $\rightarrow$ $\mathrm{MP}^\vee$ LPO $\rightarrow$ BD-N	

Constructive Reverse Mathematics

Same for WMP, FAN_Δ, BD-N, and $\mathrm{MP}^\vee$.
Same for 'mixed' theorems:

BISH (based on IL)	NSA (based on CL)
LPO $\leftrightarrow$ MP+WLPO	
MP $\leftrightarrow$ WMP + $\mathrm{MP}^\vee$	
WLPO $\rightarrow$ LLPO	
LLPO $\rightarrow$ $\mathrm{MP}^\vee$	
LPO $\rightarrow$ BD-N	
LPO $\rightarrow$ FAN_Δ	

Constructive Reverse Mathematics

Same for WMP, FAN_Δ, BD-N, and $\text{MP}^\vee$.
Same for 'mixed' theorems:

BISH (based on IL)	NSA (based on CL)
LPO $\leftrightarrow$ MP+WLPO	$\mathbb{LPO} \leftrightarrow \mathbb{MP} + \mathbb{WLPO}$
MP $\leftrightarrow$ WMP + $\text{MP}^\vee$	$\mathbb{MP} \leftrightarrow \mathbb{WMP} + \mathbb{MP}^\vee$
WLPO $\rightarrow$ LLPO	$\mathbb{WLPO} \rightarrow \mathbb{LLPO}$
LLPO $\rightarrow$ $\text{MP}^\vee$	$\mathbb{LLPO} \rightarrow \mathbb{MP}^\vee$
LPO $\rightarrow$ BD-N	$\mathbb{LPO} \rightarrow \mathbb{BD}\text{-}\mathbb{N}$
LPO $\rightarrow$ FAN_Δ	$\mathbb{LPO} \rightarrow \mathbb{FAN}_\Delta$

Conclusion

Conclusion

Reverse-engineering Reverse Mathematics (Fuchino-sensei)

Conclusion

Reverse-engineering Reverse Mathematics (Fuchino-sensei)

For Bishop's notion of algorithm, we conclude:

Algorithmic $\approx$ Ω-invariant
because
Non-algorithmic $\approx$ Non-Ω-invariant.

Why is there a connection?

Algorithmic $\approx$ Ω-invariant

because

Non-algorithmic $\approx$ Non-Ω-invariant.

Why is there a connection?

Algorithmic ≈ Ω-invariant
because
Non-algorithmic ≈ Non-Ω-invariant.

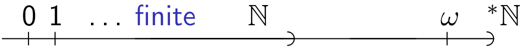
NSA

Why is there a connection?

Algorithmic ≈ Ω-invariant

because

Non-algorithmic ≈ Non-Ω-invariant.



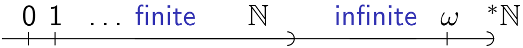
NSA

Why is there a connection?

Algorithmic $\approx$ Ω-invariant

because

Non-algorithmic $\approx$ Non-Ω-invariant.

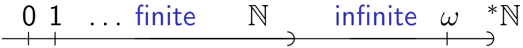


NSA

Why is there a connection?

Algorithmic $\approx$ Ω-invariant
because
Non-algorithmic $\approx$ Non-Ω-invariant.

$\mathcal{F}$ is a finite operation

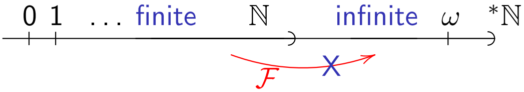


NSA

Why is there a connection?

Algorithmic $\approx$ Ω-invariant
because
Non-algorithmic $\approx$ Non-Ω-invariant.

$\mathcal{F}$ is a finite operation

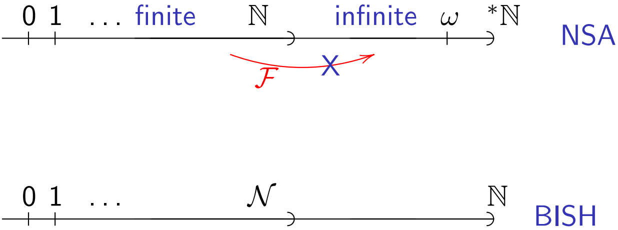


NSA

Why is there a connection?

Algorithmic ≈ Ω-invariant
because
Non-algorithmic ≈ Non-Ω-invariant.

$\mathcal{F}$ is a finite operation

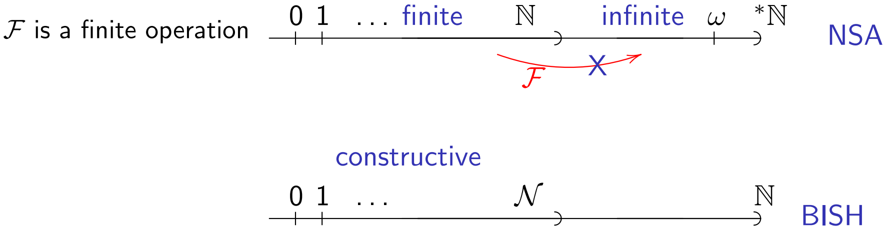


Why is there a connection?

Algorithmic $\approx$ Ω-invariant

because

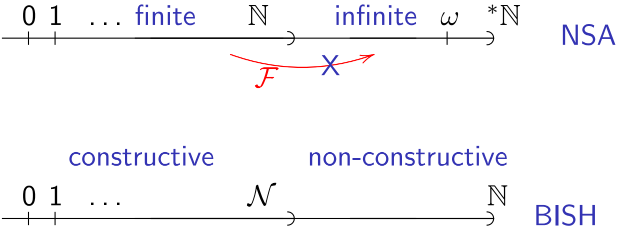
Non-algorithmic $\approx$ Non-Ω-invariant.



Why is there a connection?

Algorithmic $\approx$ Ω-invariant
because
Non-algorithmic $\approx$ Non-Ω-invariant.

$\mathcal{F}$ is a finite operation



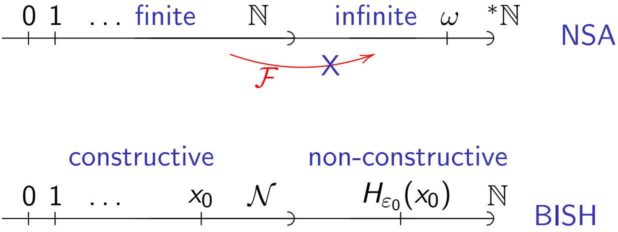
Why is there a connection?

Algorithmic $\approx$ Ω-invariant

because

Non-algorithmic $\approx$ Non-Ω-invariant.

$\mathcal{F}$ is a finite operation



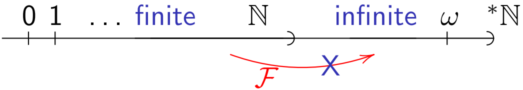
Why is there a connection?

Algorithmic $\approx$ Ω-invariant

because

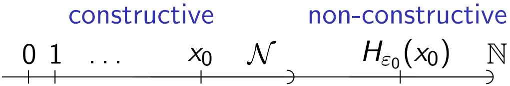
Non-algorithmic $\approx$ Non-Ω-invariant.

$\mathcal{F}$ is a finite operation



NSA

$\mathcal{C}$ is a constructive operation



BISH

Why is there a connection?

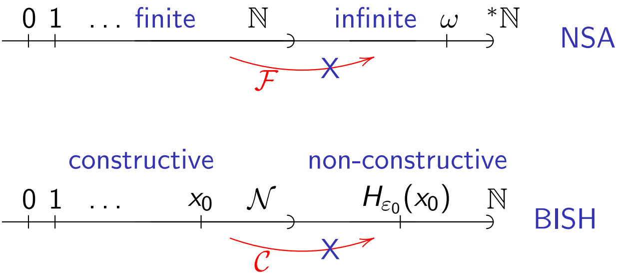
Algorithmic ≈ Ω-invariant

because

Non-algorithmic ≈ Non-Ω-invariant.

$\mathcal{F}$ is a finite operation

$\mathcal{C}$ is a constructive operation



Future research: Bounded Arithmetic

Is Ω-invariance useful for Bounded Arithmetic?

Future research: Bounded Arithmetic

Is Ω-invariance useful for Bounded Arithmetic?

1. P := PRA with bound $|f(n,x)| \leq p(|x|,|n|)$ (polynomial p).

Future research: Bounded Arithmetic

Is Ω-invariance useful for Bounded Arithmetic?

1. P := PRA with bound $|f(n,x)| \leq p(|x|,|n|)$ (polynomial p).
2. P = B := PRA without a bound, but with two sorts of variables $(\vec{x},\vec{a})$ with recursion limited to $\vec{x}$.

Future research: Bounded Arithmetic

Is Ω-invariance useful for Bounded Arithmetic?

1. P := PRA with bound $|f(n,x)| \leq p(|x|,|n|)$ (polynomial p).
2. P = B := PRA without a bound, but with two sorts of variables $(\vec{x}, \vec{a})$ with recursion limited to $\vec{x}$.
3. $\vec{x}$ are normal numbers, already constructed;
   $\vec{a}$ are safe numbers, ideal elements without a construction.

Future research: Bounded Arithmetic

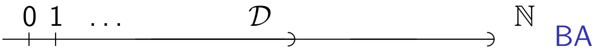
Is Ω-invariance useful for Bounded Arithmetic?

1. P := PRA with bound $|f(n, x)| \leq p(|x|, |n|)$ (polynomial p).
2. P = B := PRA without a bound, but with two sorts of variables $(\vec{x}, \vec{a})$ with recursion limited to $\vec{x}$.
3. $\vec{x}$ are normal numbers, already constructed; $\vec{a}$ are safe numbers, ideal elements without a construction.
4. The same (intuitive) picture applies:

Future research: Bounded Arithmetic

Is Ω-invariance useful for Bounded Arithmetic?

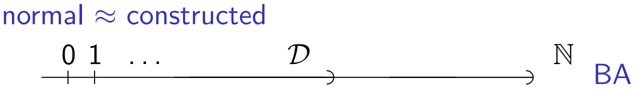
1. $P :=$ PRA with bound $|f(n,x)| \leq p(|x|,|n|)$ (polynomial p).
2. $P = B :=$ PRA without a bound, but with two sorts of variables $(\vec{x}, \vec{a})$ with recursion limited to $\vec{x}$.
3. $\vec{x}$ are normal numbers, already constructed; $\vec{a}$ are safe numbers, ideal elements without a construction.
4. The same (intuitive) picture applies:



Future research: Bounded Arithmetic

Is Ω-invariance useful for Bounded Arithmetic?

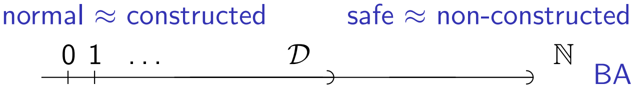
1. P := PRA with bound $|f(n,x)| \leq p(|x|,|n|)$ (polynomial p).
2. P = B := PRA without a bound, but with two sorts of variables $(\vec{x},\vec{a})$ with recursion limited to $\vec{x}$.
3. $\vec{x}$ are normal numbers, already constructed;
   $\vec{a}$ are safe numbers, ideal elements without a construction.
4. The same (intuitive) picture applies:



Future research: Bounded Arithmetic

Is Ω-invariance useful for Bounded Arithmetic?

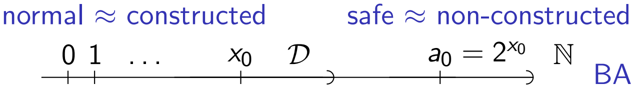
1. P := PRA with bound $|f(n,x)| \leq p(|x|,|n|)$ (polynomial p).
2. P = B := PRA without a bound, but with two sorts of variables $(\vec{x},\vec{a})$ with recursion limited to $\vec{x}$.
3. $\vec{x}$ are normal numbers, already constructed; $\vec{a}$ are safe numbers, ideal elements without a construction.
4. The same (intuitive) picture applies:



Future research: Bounded Arithmetic

Is Ω-invariance useful for Bounded Arithmetic?

1. $\mathrm{P} := \mathrm{PRA}$ with bound $|f(n,x)| \leq p(|x|,|n|)$ (polynomial p).
2. $\mathrm{P} = \mathrm{B} := \mathrm{PRA}$ without a bound, but with two sorts of variables $(\vec{x}, \vec{a})$ with recursion limited to $\vec{x}$.
3. $\vec{x}$ are normal numbers, already constructed;
   $\vec{a}$ are safe numbers, ideal elements without a construction.
4. The same (intuitive) picture applies:

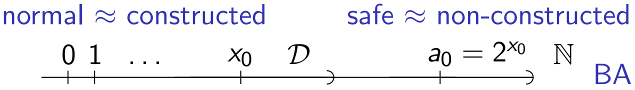


Future research: Bounded Arithmetic

Is Ω-invariance useful for Bounded Arithmetic?

1. P := PRA with bound $|f(n,x)| \leq p(|x|,|n|)$ (polynomial p).
2. P = B := PRA without a bound, but with two sorts of variables $(\vec{x}, \vec{a})$ with recursion limited to $\vec{x}$.
3. $\vec{x}$ are normal numbers, already constructed;
   $\vec{a}$ are safe numbers, ideal elements without a construction.
4. The same (intuitive) picture applies:

$\mathcal{P}$ is a polytime operation

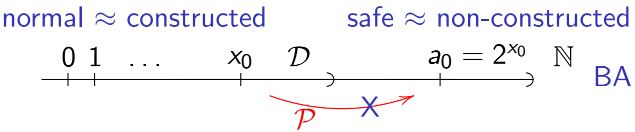


Future research: Bounded Arithmetic

Is Ω-invariance useful for Bounded Arithmetic?

1. P := PRA with bound $|f(n,x)| \leq p(|x|,|n|)$ (polynomial p).
2. P = B := PRA without a bound, but with two sorts of variables $(\vec{x}, \vec{a})$ with recursion limited to $\vec{x}$.
3. $\vec{x}$ are normal numbers, already constructed;
   $\vec{a}$ are safe numbers, ideal elements without a construction.
4. The same (intuitive) picture applies:

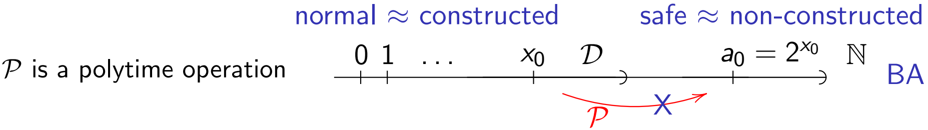
$\mathcal{P}$ is a polytime operation



Future research: Bounded Arithmetic

Is Ω-invariance useful for Bounded Arithmetic?

1. P := PRA with bound $|f(n,x)| \leq p(|x|,|n|)$ (polynomial p).
2. P = B := PRA without a bound, but with two sorts of variables $(\vec{x}, \vec{a})$ with recursion limited to $\vec{x}$.
3. $\vec{x}$ are normal numbers, already constructed; $\vec{a}$ are safe numbers, ideal elements without a construction.
4. The same (intuitive) picture applies:



P $\overset{?}{\approx}$ Safe-invariance: $(\forall \vec{x} \in \mathcal{D})(\forall \vec{a}, \vec{b} \notin \mathcal{D})\big[f(\vec{x};\vec{a}) = f(\vec{x};\vec{b})\big]$.

Future work: Type Theory

Martin-Löf intended his type theory as a foundation for BISH.

Future work: Type Theory

Martin-Löf intended his type theory as a foundation for BISH.

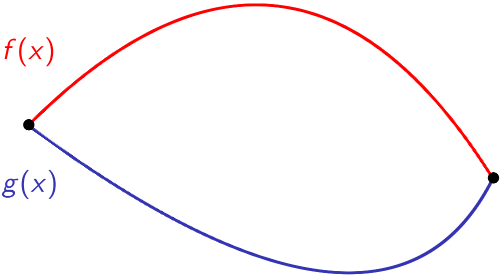
Can Ω-invariance help capture e.g. Type Theory?

Future work: Type Theory

Martin-Löf intended his type theory as a foundation for BISH.

Can Ω-invariance help capture e.g. Type Theory?

Homotopy:

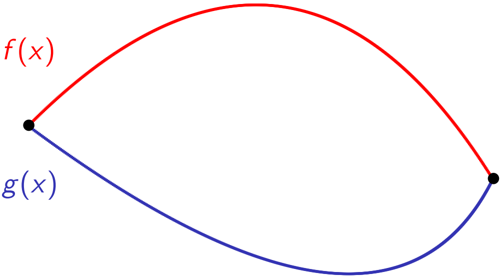


Future work: Type Theory

Martin-Löf intended his type theory as a foundation for BISH.

Can Ω-invariance help capture e.g. Type Theory?

Homotopy: continuous transformation h_t of f to g ($t \in [0,1]$).

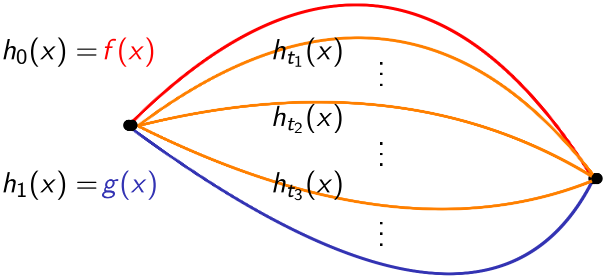


Future work: Type Theory

Martin-Löf intended his type theory as a foundation for BISH.

Can Ω-invariance help capture e.g. Type Theory?

Homotopy: continuous transformation h_t of f to g ($t \in [0,1]$).



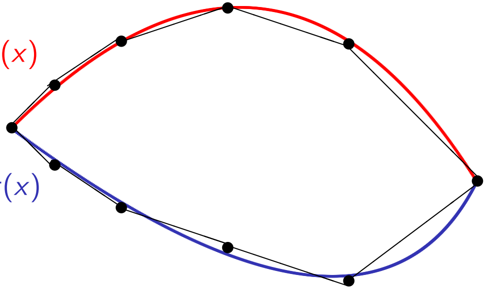
Future work: Type Theory

Martin-Löf intended his type theory as a foundation for BISH.

Can Ω-invariance help capture e.g. Type Theory?

Homotopy:

$k_\omega(x) \approx f(x)$



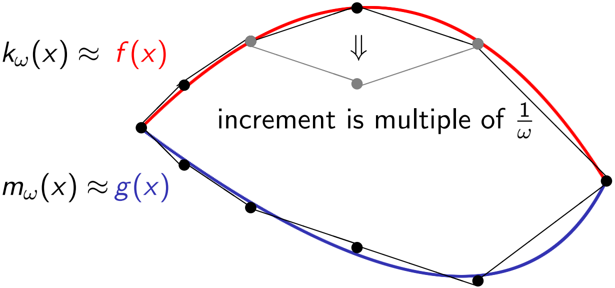
$m_\omega(x) \approx g(x)$

Future work: Type Theory

Martin-Löf intended his type theory as a foundation for BISH.

Can Ω-invariance help capture e.g. Type Theory?

Homotopy:

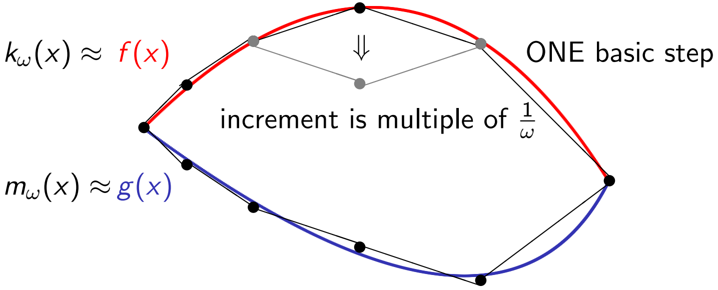


Future work: Type Theory

Martin-Löf intended his type theory as a foundation for BISH.

Can Ω-invariance help capture e.g. Type Theory?

Homotopy:

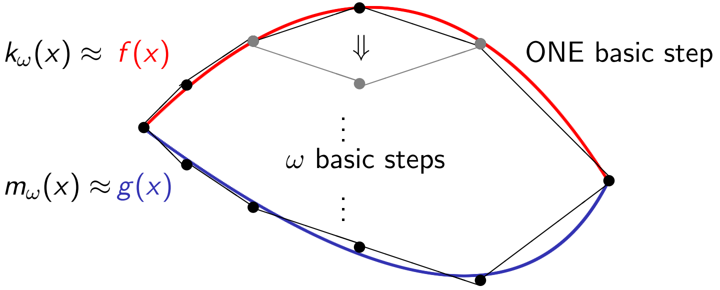


Future work: Type Theory

Martin-Löf intended his type theory as a foundation for BISH.

Can Ω-invariance help capture e.g. Type Theory?

Homotopy:

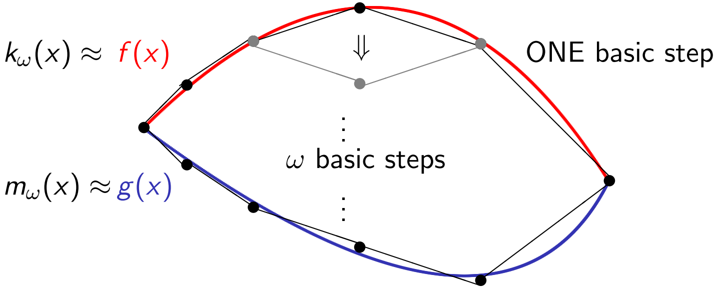


Future work: Type Theory

Martin-Löf intended his type theory as a foundation for BISH.

Can Ω-invariance help capture e.g. Type Theory?

Homotopy:



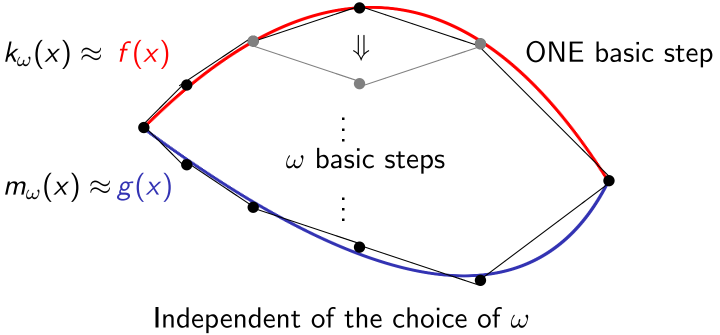
Independent of the choice of ω

Future work: Type Theory

Martin-Löf intended his type theory as a foundation for BISH.

Can Ω-invariance help capture e.g. Type Theory?

Homotopy: $\approx$ Ω-invariant broken-line transformation $h_{\omega,t}$ of f to g.



On (anti)realism

In the right framework, Ω-invariance captures Bishop's notion algorithm indirectly and from the outside.

On (anti)realism

In the right framework, Ω-invariance captures Bishop's notion algorithm indirectly and from the outside.

An analogy:

On (anti)realism

In the right framework, Ω-invariance captures Bishop's notion algorithm indirectly and from the outside.

An analogy:

Brains in vats
VERSUS

On (anti)realism

In the right framework, Ω-invariance captures Bishop's notion algorithm indirectly and from the outside.

An analogy:

Brains in vats
VERSUS
Intuitionists/constructivists in NSA.

On (anti)realism

In the right framework, Ω-invariance captures Bishop's notion algorithm indirectly and from the outside.

An analogy:

Brains in vats
VERSUS
Intuitionists/constructivists in NSA.

How does X know he is not actually sitting in Y?

Philosophy of Physics

Philosophy of Physics

Why is Mathematics in Physics so constructive/computable?

Philosophy of Physics

Why is Mathematics in Physics so constructive/computable?

Indeed, in Physics, calculations are explicit and existence statements come with a construction (symbolically or numerically).

Philosophy of Physics

Why is Mathematics in Physics so constructive/computable?

Indeed, in Physics, calculations are explicit and existence statements come with a construction (symbolically or numerically).

The answer: in Physics, an informal version of NSA is used to date. (The notorious ‘epsilon-delta’ method was never adopted).

Philosophy of Physics

Why is Mathematics in Physics so constructive/computable?

Indeed, in Physics, calculations are explicit and existence statements come with a construction (symbolically or numerically).

The answer: in Physics, an informal version of NSA is used to date. (The notorious 'epsilon-delta' method was never adopted).

Also, in Physics, the end result of a calculation should have physical meaning (modeling of reality).

Philosophy of Physics

Why is Mathematics in Physics so constructive/computable?

Indeed, in Physics, calculations are explicit and existence statements come with a construction (symbolically or numerically).

The answer: in Physics, an informal version of NSA is used to date. (The notorious 'epsilon-delta' method was never adopted).

Also, in Physics, the end result of a calculation should have physical meaning (modeling of reality).

An end result with physical meaning will not depend on the infinite number/infinitesimal used, i.e. it is Ω-invariant. (Connes)

On Robustness

Robustness = invariance under variation of parameters.

On Robustness

Robustness = invariance under variation of parameters.

Parts of Reverse Mathematics and Computability are robust.

On Robustness

Robustness $=$ invariance under variation of parameters.

Parts of Reverse Mathematics and Computability are robust.

Robustness is a mixture of syntax and semantics.

On Robustness

Robustness = invariance under variation of parameters.

Parts of Reverse Mathematics and Computability are robust.

Robustness is a mixture of syntax and semantics. (Ontology of mathematics?)

On Robustness

Robustness = invariance under variation of parameters.

Parts of Reverse Mathematics and Computability are robust.

Robustness is a mixture of syntax and semantics. (Ontology of mathematics?)

Robustness (for scientific models) is a no-false-positives guarantee.

On Robustness

Robustness = invariance under variation of parameters.

Parts of Reverse Mathematics and Computability are robust.

Robustness is a mixture of syntax and semantics. (Ontology of mathematics?)

Robustness (for scientific models) is a no-false-positives guarantee.

A distant dream: To provide a framework for building scientific models that come with a proof of robustness,

On Robustness

Robustness = invariance under variation of parameters.

Parts of Reverse Mathematics and Computability are robust.

Robustness is a mixture of syntax and semantics. (Ontology of mathematics?)

Robustness (for scientific models) is a no-false-positives guarantee.

A distant dream: To provide a framework for building scientific models that come with a proof of robustness, in the same way as Type Theory provides a framework of building computer programs that come with a proof of correctness.

Final Thoughts

Final Thoughts

And what are these [infinitesimals]? [...] They are neither finite Quantities nor Quantities infinitely small, nor yet nothing. May we not call them the ghosts of departed quantities?

George Berkeley, *The Analyst*

Final Thoughts

And what are these [infinitesimals]? [. . .] They are neither finite Quantities nor Quantities infinitely small, nor yet nothing. May we not call them the ghosts of departed quantities?

George Berkeley, *The Analyst*

...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

And what are these [infinitesimals]? [...] They are neither finite Quantities nor Quantities infinitely small, nor yet nothing. May we not call them the ghosts of departed quantities?
George Berkeley, *The Analyst*

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

We thank the John Templeton Foundation for its generous support!

Final Thoughts

And what are these [infinitesimals]? [...] They are neither finite Quantities nor Quantities infinitely small, nor yet nothing. May we not call them the ghosts of departed quantities?
George Berkeley, *The Analyst*

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

We thank the John Templeton Foundation for its generous support!

Thank you for your attention!

Final Thoughts

And what are these [infinitesimals]? [...] They are neither finite Quantities nor Quantities infinitely small, nor yet nothing. May we not call them the ghosts of departed quantities?

George Berkeley, *The Analyst*

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

Kurt Gödel

We thank the John Templeton Foundation for its generous support!

Thank you for your attention!

Any questions?