Nonstandard Analysis: A New Way to Compute

Sam Sanders[1]

Midwest Computability Seminar, Nov. 15, 2012

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

How to compute in NSA: Ω-invariance

How to compute in NSA: Ω-invariance

Definition (Ω-invariance)

For $\psi(n, m) \in \Delta_0$ and $\omega \in \Omega$, the formula $\psi(n, \omega)$ is Ω-invariant if

How to compute in NSA: Ω-invariance

Definition (Ω-invariance)

For $\psi(n, m) \in \Delta_0$ and $\omega \in \Omega$, the formula $\psi(n, \omega)$ is Ω-invariant if

$$(\forall n \in \mathbb{N})(\forall \omega' \in \Omega)[\psi(n, \omega) \leftrightarrow \psi(n, \omega')].$$

How to compute in NSA: Ω-invariance

Definition (Ω-invariance)

For $\psi(n,m) \in \Delta_0$ and $\omega \in \Omega$, the formula $\psi(n,\omega)$ is Ω-invariant if

$$(\forall n \in \mathbb{N})(\forall \omega' \in \Omega)[\psi(n,\omega) \leftrightarrow \psi(n,\omega')].$$

Note that $\psi(n,\omega)$ is independent of the choice of $\omega \in \Omega$.

How to compute in NSA: Ω-invariance

Definition (Ω-invariance)

For $\psi(n, m) \in \Delta_0$ and $\omega \in \Omega$, the formula $\psi(n, \omega)$ is Ω-invariant if

$$(\forall n \in \mathbb{N})(\forall \omega' \in \Omega)[\psi(n, \omega) \leftrightarrow \psi(n, \omega')].$$

Note that $\psi(n, \omega)$ is independent of the choice of $\omega \in \Omega$.

Principle (Ω-CA)

For all Ω-invariant $\psi(n, \omega)$, we have

$$(\exists X^s \subset \mathbb{N})(\forall n \in \mathbb{N})(n \in X^s \leftrightarrow \psi(n, \omega)).$$

How to compute in NSA: Ω-invariance

Definition (Ω-invariance)

For $\psi(n,m) \in \Delta_0$ and $\omega \in \Omega$, the formula $\psi(n,\omega)$ is Ω-invariant if

$$(\forall n \in \mathbb{N})(\forall \omega' \in \Omega)[\psi(n,\omega) \leftrightarrow \psi(n,\omega')].$$

Note that $\psi(n,\omega)$ is independent of the choice of $\omega \in \Omega$.

Principle (Ω-CA)

For all Ω-invariant $\psi(n,\omega)$, we have

$$(\exists X^s \subset \mathbb{N})(\forall n \in \mathbb{N})(n \in X^s \leftrightarrow \psi(n,\omega)).$$

Theorem

Ω-CA *implies* Δ_1^0-CA (*Turing comp.* $\subseteq$ Ω-*invariance*)

How to compute in NSA: Ω-invariance

Definition (Ω-invariance)

For $\psi(n, m) \in \Delta_0$ and $\omega \in \Omega$, the formula $\psi(n, \omega)$ is Ω-invariant if

$$(\forall n \in \mathbb{N})(\forall \omega' \in \Omega)[\psi(n, \omega) \leftrightarrow \psi(n, \omega')].$$

Note that $\psi(n, \omega)$ is independent of the choice of $\omega \in \Omega$.

Principle (Ω-CA)

For all Ω-invariant $\psi(n, \omega)$, we have

$$(\exists X^s \subset \mathbb{N})(\forall n \in \mathbb{N})(n \in X^s \leftrightarrow \psi(n, \omega)).$$

Theorem

Ω-CA *implies* Δ^0_1-CA (*Turing comp.* $\subseteq$ *Ω-invariance*)

$^*RCA_0 + \Omega$-CA $\equiv_{cons}$ RCA_0 '(*Ω-invariance $\subseteq$ Turing comp.*)'

How to compute in NSA: Ω-invariance

Definition (Ω-invariance)

For $\psi(n,m) \in \Delta_0$ and $\omega \in \Omega$, the formula $\psi(n,\omega)$ is Ω-invariant if

$$(\forall n \in \mathbb{N})(\forall \omega' \in \Omega)[\psi(n,\omega) \leftrightarrow \psi(n,\omega')].$$

Note that $\psi(n,\omega)$ is independent of the choice of $\omega \in \Omega$.

Principle (Ω-CA)

For all Ω-invariant $\psi(n,\omega)$, we have

$$(\exists X^s \subset \mathbb{N})(\forall n \in \mathbb{N})(n \in X^s \leftrightarrow \psi(n,\omega)).$$

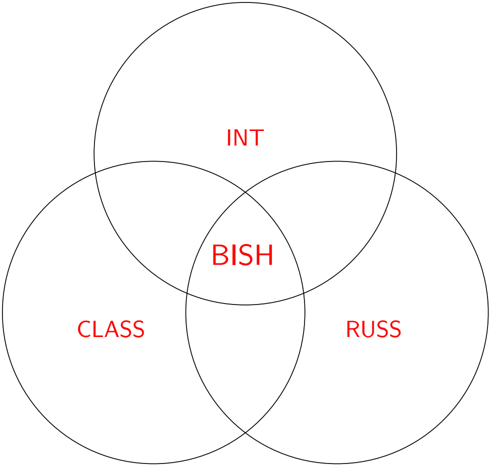
Theorem

Ω-CA *implies* Δ_1^0-CA (*Turing comp.* $\subseteq$ *Ω-invariance*)

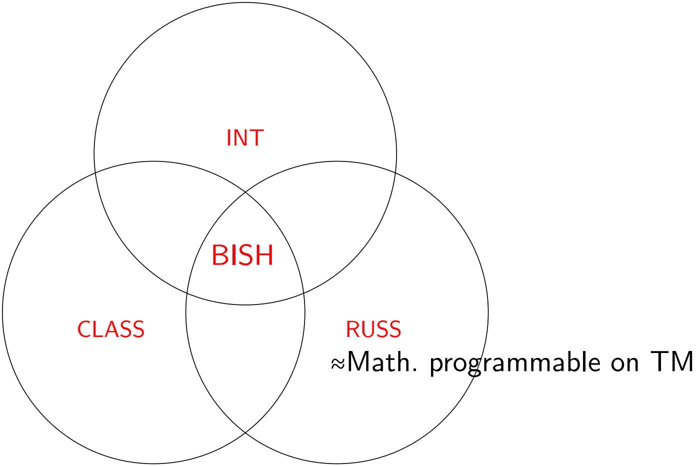
${}^*\mathrm{RCA}_0 + \Omega\text{-CA} \equiv_{\mathrm{cons}} \mathrm{RCA}_0$ ‘(*Ω-invariance* $\subseteq$ *Turing comp.*)’

(Physics / TCS)

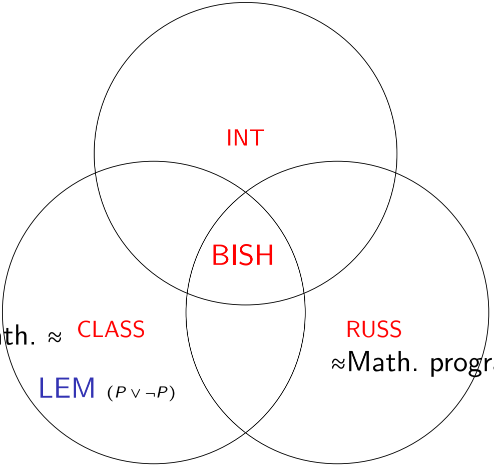
Son of a...



Son of a. . .



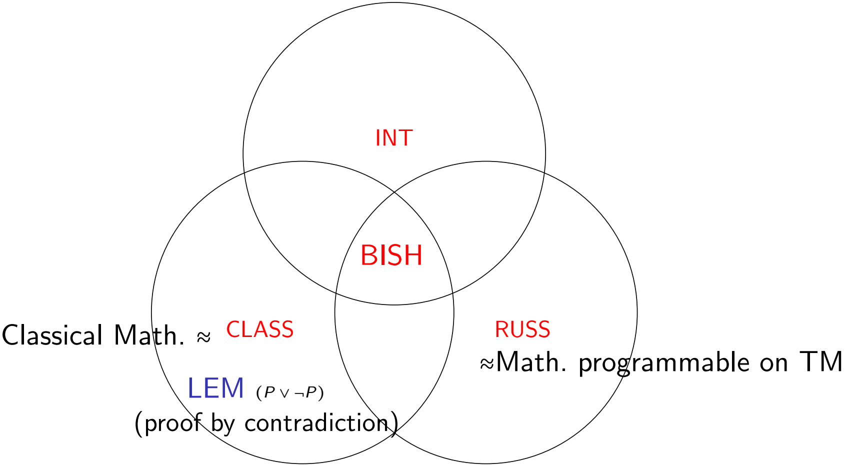
Son of a...



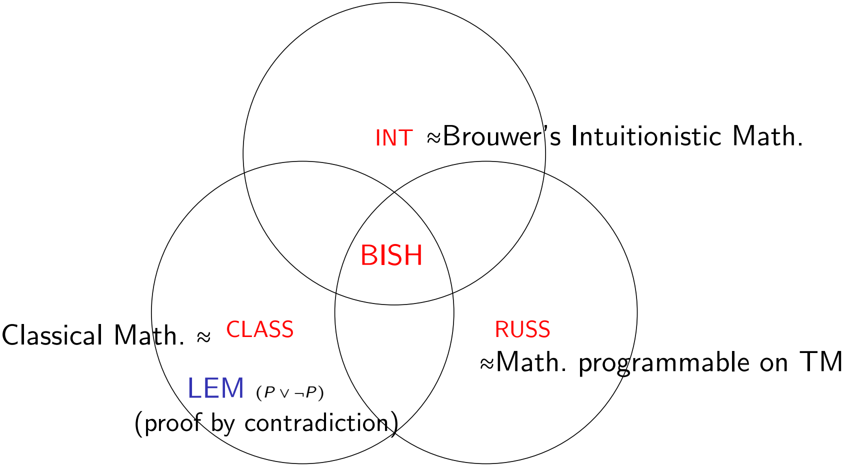
Classical Math. ≈ CLASS

RUSS ≈Math. programmable on TM

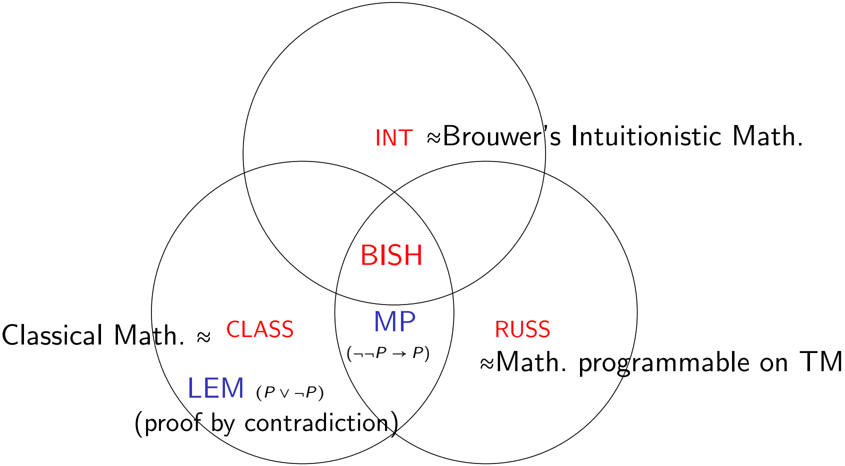
Son of a. . .


INT
BISH
Classical Math. ≈
CLASS
RUSS
≈Math. programmable on TM
LEM $(P \vee \neg P)$
(proof by contradiction)

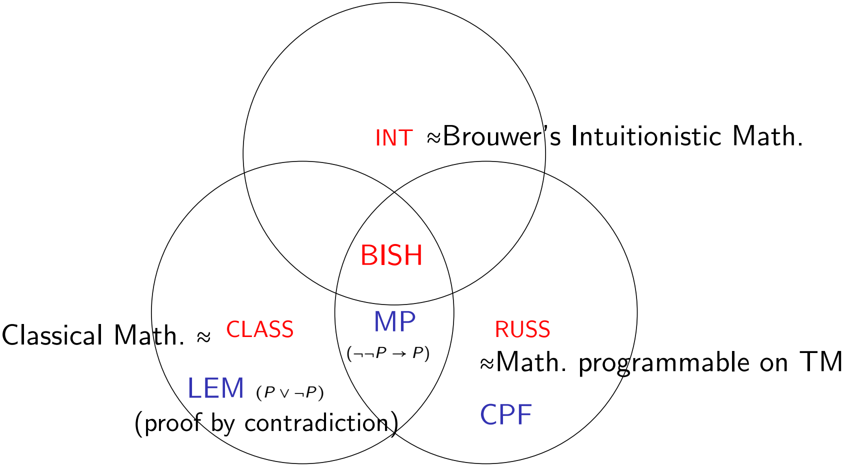
Son of a. . .


INT ≈Brouwer's Intuitionistic Math.
BISH
Classical Math. ≈ CLASS
RUSS
≈Math. programmable on TM
LEM $(P \vee \neg P)$
(proof by contradiction)

Son of a. . .

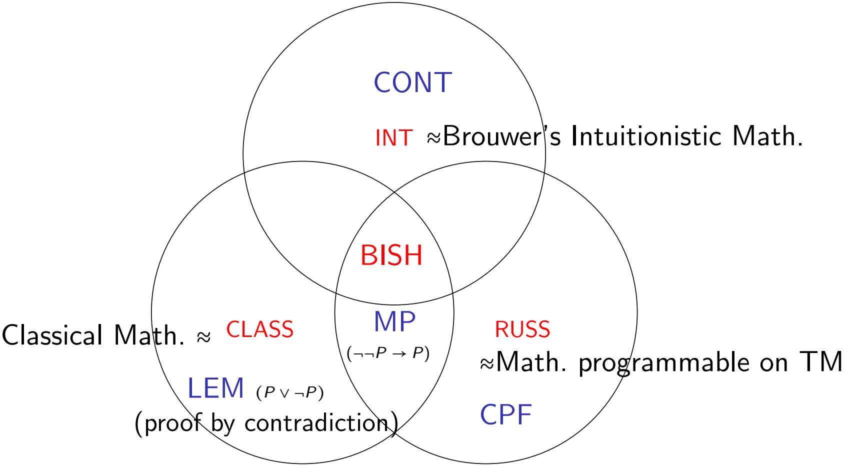

INT ≈Brouwer's Intuitionistic Math.
BISH
Classical Math. ≈ CLASS
MP
(¬¬P → P)
RUSS
≈Math. programmable on TM
LEM (P ∨ ¬P)
(proof by contradiction)

Son of a...

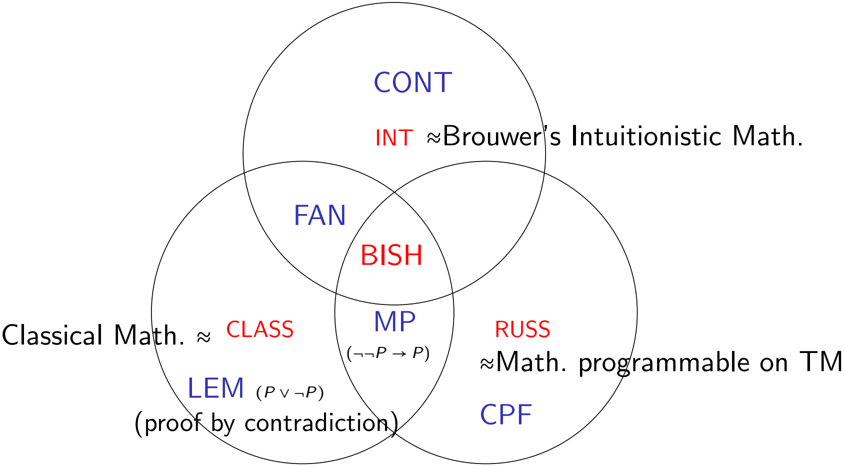


INT ≈Brouwer's Intuitionistic Math.
BISH
Classical Math. ≈ CLASS
MP
$(\neg\neg P \rightarrow P)$
RUSS
≈Math. programmable on TM
LEM $(P \vee \neg P)$
(proof by contradiction)
CPF

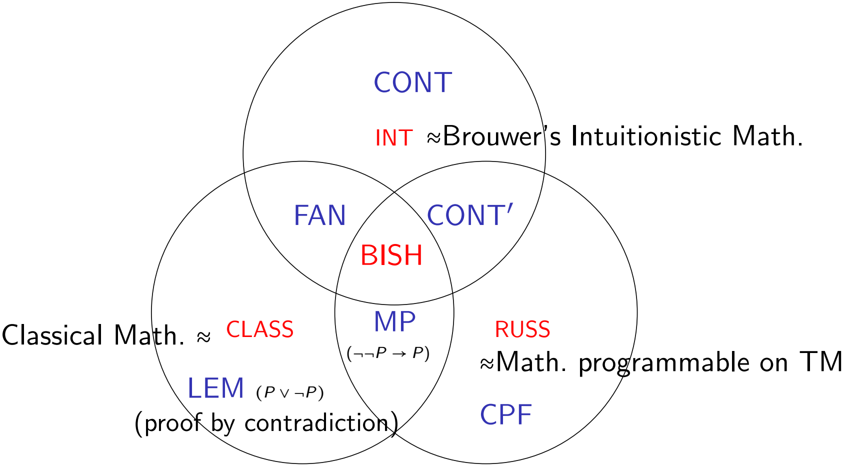
Son of a. . .


CONT
INT ≈Brouwer's Intuitionistic Math.
BISH
Classical Math. ≈ CLASS
MP
(¬¬P → P)
RUSS
≈Math. programmable on TM
LEM (P ∨ ¬P)
(proof by contradiction)
CPF

Son of a. . .

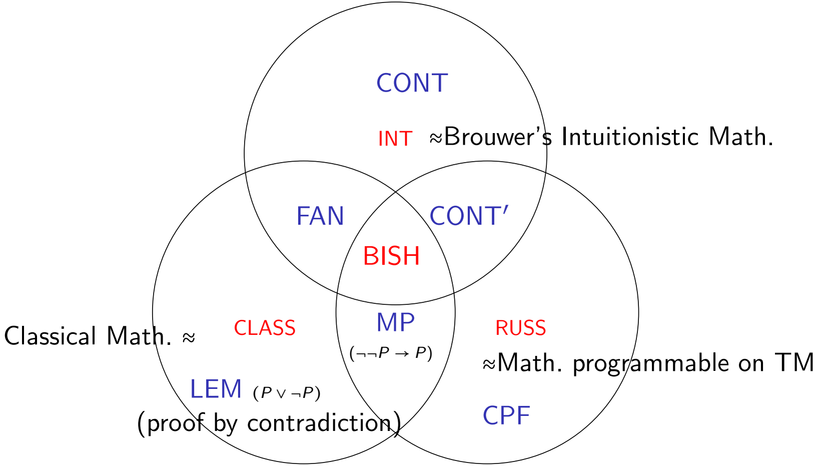

CONT
INT ≈Brouwer's Intuitionistic Math.
FAN
BISH
Classical Math. ≈ CLASS
MP
(¬¬P → P)
RUSS
≈Math. programmable on TM
LEM (P ∨ ¬P)
(proof by contradiction)
CPF

Son of a...


CONT
INT ≈Brouwer's Intuitionistic Math.
FAN
CONT′
BISH
Classical Math. ≈ CLASS
MP
(¬¬P → P)
RUSS
≈Math. programmable on TM
LEM (P ∨ ¬P)
(proof by contradiction)
CPF

Son of a...

Errett Bishop's Constructive Analysis (also 'BISH') is a constructive redevelopment of Mathematics, consistent with CLASS, RUSS and INT.



Algorithm and Proof in Constructive Analysis

Errett Bishop's Constructive Analysis (BISH) is a constructive redevelopment of Mathematics, where algorithm and proof are central.

Algorithm and Proof in Constructive Analysis

Errett Bishop's Constructive Analysis (BISH) is a constructive redevelopment of Mathematics, where algorithm and proof are central.

Definition (Logical connectives in BISH: BHK)

Algorithm and Proof in Constructive Analysis

Errett Bishop's Constructive Analysis (BISH) is a constructive redevelopment of Mathematics, where algorithm and proof are central.

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.

Algorithm and Proof in Constructive Analysis

Errett Bishop's Constructive Analysis (BISH) is a constructive redevelopment of Mathematics, where algorithm and proof are central.

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.

Algorithm and Proof in Constructive Analysis

Errett Bishop's Constructive Analysis (BISH) is a constructive redevelopment of Mathematics, where algorithm and proof are central.

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.

Algorithm and Proof in Constructive Analysis

Errett Bishop's Constructive Analysis (BISH) is a constructive redevelopment of Mathematics, where algorithm and proof are central.

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. $\neg P \equiv P \to (0 = 1)$.

Algorithm and Proof in Constructive Analysis

Errett Bishop's Constructive Analysis (BISH) is a constructive redevelopment of Mathematics, where algorithm and proof are central.

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 $P(x_0)$

Algorithm and Proof in Constructive Analysis

Errett Bishop's Constructive Analysis (BISH) is a constructive redevelopment of Mathematics, where algorithm and proof are central.

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, $x \in A \to P(x)$.

Lost in translation

Lost in translation

BISH (based on BHK)

Lost in translation

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)

Lost in translation

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)
Central: algorithm and proof	

Lost in translation

BISH (based on BHK)

Central: algorithm and proof

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

$\mathbb{NSA}$ (based on CL)

Lost in translation

BISH (based on BHK)

Central: algorithm and proof

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

$\mathbb{NSA}$ (based on CL)

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

Lost in translation

BISH (based on BHK)

Central: algorithm and proof

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

$\mathbb{NSA}$ (based on CL)

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

$A \mathbb{V} B$:

Lost in translation

BISH (based on BHK)

Central: algorithm and proof

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

$\mathbb{NSA}$ (based on CL)

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

$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x},\omega)$ s.t.

$$\psi(\vec{x},\omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]]$$
$$\wedge$$
$$\neg\psi(\vec{x},\omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

Lost in translation

BISH (based on BHK)

Central: algorithm and proof

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

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

$\mathbb{NSA}$ (based on CL)

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

$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x},\omega)$ s.t.

$$\psi(\vec{x},\omega) \rightarrow [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]]$$
$$\wedge$$
$$\neg\psi(\vec{x},\omega) \rightarrow [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

Lost in translation

BISH (based on BHK)

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

$\mathbb{NSA}$ (based on CL)

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

$A \,\mathbb{V}\, B$: There is Ω-invariant $\psi(\vec{x},\omega)$ s.t.

$$\psi(\vec{x},\omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]]$$

$$\wedge$$

$$\neg\psi(\vec{x},\omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

$A \Rightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \to \big[B \wedge [B \in \mathbb{T}]\big]$

Lost in translation

BISH (based on BHK)

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

$\mathbb{NSA}$ (based on CL)

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

$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x}, \omega)$ s.t.

$$\psi(\vec{x}, \omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]]$$
$$\wedge$$
$$\neg\psi(\vec{x}, \omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

$A \Rrightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \to \big[B \wedge [B \in \mathbb{T}]\big]$

'$A \in \mathbb{T}$' means 'A satisfies Transfer'.

Lost in translation

BISH (based on BHK)

Central: algorithm and proof

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

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

$\mathbb{NSA}$ (based on CL)

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

$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x},\omega)$ s.t.

$$\psi(\vec{x},\omega) \rightarrow [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]]$$
$$\wedge$$
$$\neg\psi(\vec{x},\omega) \rightarrow [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

$A \Rightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \rightarrow \big[B \wedge [B \in \mathbb{T}]\big]$

'$A \in \mathbb{T}$' means 'A satisfies Transfer'.

E.g. '$(\forall n \in \mathbb{N})\varphi(n) \in \mathbb{T}$' is $[(\forall n \in \mathbb{N})\varphi(n) \rightarrow (\forall n \in {}^*\mathbb{N})\varphi(n)]$

Lost in translation

BISH (based on BHK) | $\mathbb{NSA}$ (based on CL)

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

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

$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x},\omega)$ s.t.

$$\psi(\vec{x},\omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]] \wedge \neg\psi(\vec{x},\omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

$A \Rightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \to \big[B \wedge [B \in \mathbb{T}]\big]$

'$A \in \mathbb{T}$' 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 BHK)

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

$\neg A$: $A \to (0 = 1)$

$\mathbb{NSA}$ (based on CL)

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

$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x},\omega)$ s.t.

$$\psi(\vec{x},\omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]]$$
$$\wedge$$
$$\neg\psi(\vec{x},\omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

$A \Rrightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \to \big[B \wedge [B \in \mathbb{T}]\big]$

Lost in translation

BISH (based on BHK)

Central: algorithm and proof

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

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

$\neg A$: $A \rightarrow (0 = 1)$

$\mathbb{NSA}$ (based on CL)

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

$A \mathbin{\mathbb{V}} B$: There is Ω-invariant $\psi(\vec{x}, \omega)$ s.t.

$$\psi(\vec{x}, \omega) \rightarrow [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]]$$
$$\wedge$$
$$\neg\psi(\vec{x}, \omega) \rightarrow [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

$A \Rightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \rightarrow \big[B \wedge [B \in \mathbb{T}]\big]$

$\sim A$: $A \Rightarrow (0 = 1)$

Lost in translation

BISH (based on BHK)

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

$\neg A$: $A \to (0 = 1)$

$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$

$\mathbb{NSA}$ (based on CL)

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

$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x}, \omega)$ s.t.

$$\psi(\vec{x},\omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]] \\ \wedge \\ \neg\psi(\vec{x},\omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

$A \Rrightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \to \big[B \wedge [B \in \mathbb{T}]\big]$

$\sim A$: $A \Rrightarrow (0 = 1)$

Lost in translation

BISH (based on BHK)

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

$\neg A$: $A \to (0 = 1)$

$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$

$\mathbb{NSA}$ (based on CL)

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

$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x},\omega)$ s.t.

$$\psi(\vec{x},\omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]] \\ \wedge \\ \neg\psi(\vec{x},\omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

$A \Rrightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \to \big[B \wedge [B \in \mathbb{T}]\big]$

$\sim A$: $A \Rrightarrow (0 = 1)$

$(\exists x)A(x)$: an Ω-inv. proc. computes x_0 such that $A(x_0)$

Lost in translation

BISH (based on BHK)

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

$\neg A$: $A \to (0 = 1)$

$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$

$\mathbb{NSA}$ (based on CL)

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

$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x},\omega)$ s.t.

$$\psi(\vec{x},\omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]]$$
$$\wedge$$
$$\neg\psi(\vec{x},\omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

$A \Rightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \to \big[B \wedge [B \in \mathbb{T}]\big]$

$\sim A$: $A \Rightarrow (0 = 1)$

$(\exists x)A(x)$: 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 BHK)

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

$\neg A$: $A \to (0 = 1)$

$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$

$\mathbb{NSA}$ (based on CL)

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

$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x}, \omega)$ s.t.

$$\psi(\vec{x},\omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]]$$
$$\wedge$$
$$\neg\psi(\vec{x},\omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

$A \Rrightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \to \big[B \wedge [B \in \mathbb{T}]\big]$

$\sim A$: $A \Rrightarrow (0 = 1)$

$(\exists x)A(x)$: an Ω-inv. proc. computes x_0 such that $A(x_0)$

$$\sim[(\forall n \in \mathbb{N})A(n)] \equiv (\exists n \in \mathbb{N}_1)\sim A(n)$$

WEAKER than $(\exists n \in \mathbb{N})\sim A(n)$.

Lost in translation

BISH (based on BHK)

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

$\neg A$: $A \to (0 = 1)$

$(\exists x)A(x)$: an algo computes x_0 such that $A(x_0)$

$\neg[(\forall n \in \mathbb{N})A(n)]$ is WEAKER than $(\exists n \in \mathbb{N})\neg A(n)$.

$\mathbb{NSA}$ (based on CL)

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

$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x}, \omega)$ s.t.

$$\psi(\vec{x}, \omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]] \\ \wedge \\ \neg\psi(\vec{x}, \omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$$

$A \Rrightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \to \big[B \wedge [B \in \mathbb{T}]\big]$

$\sim A$: $A \Rrightarrow (0 = 1)$

$(\exists x)A(x)$: an Ω-inv. proc. computes x_0 such that $A(x_0)$

$\sim[(\forall n \in \mathbb{N})A(n)] \equiv (\exists n \in \mathbb{N}_1)\sim A(n)$

WEAKER than $(\exists n \in \mathbb{N})\sim A(n)$.

Lost in translation

BISH (based on BHK)	$\mathbb{NSA}$ (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	$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x},\omega)$ s.t. $\psi(\vec{x},\omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]]$ $\wedge$ $\neg\psi(\vec{x},\omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \to \big[B \wedge [B \in \mathbb{T}]\big]$
$\neg A$: $A \to (0 = 1)$	$\sim A$: $A \Rightarrow (0 = 1)$
$(\exists x)A(x)$: 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 BHK)	$\mathbb{NSA}$ (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	$A \mathbb{V} B$: There is Ω-invariant $\psi(\vec{x},\omega)$ s.t. $\psi(\vec{x},\omega) \to [A(\vec{x}) \wedge [A(\vec{x}) \in \mathbb{T}]]$ $\wedge$ $\neg\psi(\vec{x},\omega) \to [B(\vec{x}) \wedge [B(\vec{x}) \in \mathbb{T}]]$
$A \to B$: an algo converts a proof of A to a proof of B	$A \Rightarrow B$: $\big[A \wedge [A \in \mathbb{T}]\big] \to \big[B \wedge [B \in \mathbb{T}]\big]$
$\neg A$: $A \to (0 = 1)$	$\sim A$: $A \Rightarrow (0 = 1)$
$(\exists x)A(x)$: 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 it preserves all essential features of BISH (e.g. CRM)

Constructive Reverse Mathematics

Constructive Reverse Mathematics

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)
non-constructive/non-algorithmic	

Constructive Reverse Mathematics

BISH (based on BHK)

non-constructive/non-algorithmic

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

$\updownarrow$

$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics

BISH (based on BHK)

non-constructive/non-algorithmic

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

$\updownarrow$

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

$\updownarrow$

$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics

BISH (based on BHK) | $\mathbb{NSA}$ (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 BHK) | $\mathbb{NSA}$ (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 BHK)	$\mathbb{NSA}$ (based on CL)
non-constructive/non-algorithmic	Transfer Principle
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 BHK)

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

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} \sim P$

$\Longleftrightarrow$

Constructive Reverse Mathematics

BISH (based on BHK)

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

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} \sim P$

$\Longleftrightarrow$

$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbb{V} \sim(x > 0))$

$\Longleftrightarrow$

Constructive Reverse Mathematics

BISH (based on BHK)

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

$\Longleftrightarrow$

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} \sim P$

$\Longleftrightarrow$

$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbb{V} \sim(x > 0))$

$\Longleftrightarrow$

$\mathbb{MCT}$: monotone convergence thm

Constructive Reverse Mathematics

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

Constructive Reverse Mathematics

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

Constructive Reverse Mathematics

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

Constructive Reverse Mathematics

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)
non-constructive/non-algorithmic	Transfer Principle
LPO: For $P \in \Sigma_1$, $P \vee \neg P$	$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} \sim P$
$\updownarrow$	$\Longleftrightarrow$
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$	$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbb{V} \sim(x > 0))$
$\updownarrow$	$\Longleftrightarrow$
MCT: monotone convergence thm	$\mathbb{MCT}$: monotone convergence thm
$\updownarrow$ (limit computed by algo) $\Longleftrightarrow$	(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 BHK)

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

$\updownarrow$ (limit computed by algo)

CIT: Cantor intersection thm

(point in intersection computed by algo)

$\Longleftrightarrow$

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} \sim P$

$\Longleftrightarrow$

$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbb{V} \sim(x > 0))$

$\Longleftrightarrow$

$\mathbb{MCT}$: monotone convergence thm

(limit computed by Ω-inv. proc.)

$\mathbb{CIT}$: Cantor intersection thm

(point in intersection computed by Ω-inv. proc.)

Constructive Reverse Mathematics

BISH (based on BHK)		$\mathbb{NSA}$ (based on CL)
non-constructive/non-algorithmic		Transfer Principle
LPO: For $P \in \Sigma_1$, $P \vee \neg P$		$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} \sim P$
$\updownarrow$		$\Longleftrightarrow$
LPR: $(\forall x \in \mathbb{R})(x > 0 \vee \neg(x > 0))$		$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \mathbb{V} \sim (x > 0))$
$\updownarrow$		$\Longleftrightarrow$
MCT: monotone convergence thm		$\mathbb{MCT}$: monotone convergence thm
$\updownarrow$ (limit computed by algo)	$\Longleftrightarrow$	(limit computed by Ω-inv. proc.)
CIT: Cantor intersection thm		$\mathbb{CIT}$: Cantor intersection thm
		$\Longleftrightarrow$
		Π_1-TRANS^{SET}
		$(\forall n \in \mathbb{N})\varphi(n, \vec{X}) \to (\forall n \in {}^*\mathbb{N})\varphi(n, {}^*\vec{X})$

Constructive Reverse Mathematics

BISH (based on BHK)

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

$\updownarrow$ (limit computed by algo)

CIT: Cantor intersection thm

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LPO}$: For $P \in \Sigma_1$, $P \mathbb{V} {\sim} P$

$\Longleftrightarrow$

$\mathbb{LPR}$: $(\forall x \in \mathbb{R})(x > 0 \,\mathbb{V} {\sim}(x > 0))$

$\Longleftrightarrow$

$\mathbb{MCT}$: monotone convergence thm

$\Longleftrightarrow$ (limit computed by Ω-inv. proc.)

$\mathbb{CIT}$: Cantor intersection thm

$\Longleftrightarrow$

Π_1-TRANS^{SET}

$(\forall n \in \mathbb{N})\varphi(n, \vec{X}) \to (\forall n \in {}^*\mathbb{N})\varphi(n, {}^*\vec{X})$

$\mathbb{NSA}$ does prove $(\forall \delta \in \mathbb{R})\big[\delta > 0 \Rightarrow (x > 0) \,\mathbb{V} (x < \delta)\big]$.

BISH does prove $(\forall \delta \in \mathbb{R})\big[\delta > 0 \to (x > 0) \vee (x < \delta)\big]$.

Constructive Reverse Mathematics II

Constructive Reverse Mathematics II

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)
non-constructive/non-algorithmic	

Constructive Reverse Mathematics II

BISH (based on BHK)

non-constructive/non-algorithmic

LLPO

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

$\updownarrow$

$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics II

BISH (based on BHK)

non-constructive/non-algorithmic

LLPO

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

$\updownarrow$

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

$\updownarrow$

$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics II

BISH (based on BHK)

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

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

$\updownarrow$

$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics II

BISH (based on BHK)

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

$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics II

BISH (based on BHK)

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

$\mathbb{NSA}$ (based on CL)

Transfer Principle

Constructive Reverse Mathematics II

BISH (based on BHK)

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

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LLPO}$

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

$\Longleftrightarrow$

Constructive Reverse Mathematics II

BISH (based on BHK)

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

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LLPO}$

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

$\Longleftrightarrow$

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

$\Longleftrightarrow$

Constructive Reverse Mathematics II

BISH (based on BHK)

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

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LLPO}$

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

$\Longleftrightarrow$

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

$\Longleftrightarrow$

$\mathbb{NIL}$

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

$\Longleftrightarrow$

Constructive Reverse Mathematics II

BISH (based on BHK)

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

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LLPO}$

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

$\Longleftrightarrow$

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

$\Longleftrightarrow$

$\mathbb{NIL}$

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

$\Longleftrightarrow$

$\mathbb{IVT}$: Intermediate value theorem

Constructive Reverse Mathematics II

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)
non-constructive/non-algorithmic	Transfer Principle
LLPO	$\mathbb{LLPO}$
For $P, Q \in \Sigma_1$, $\neg(P \wedge Q) \rightarrow \neg P \vee \neg Q$	For $P, Q \in \Sigma_1$, $\sim(P \wedge Q) \Rightarrow \sim P \mathbb{V} \sim Q$
$\updownarrow$	$\Longleftrightarrow$
LLPR: $(\forall x \in \mathbb{R})(x \geq 0 \vee x \leq 0)$	$\mathbb{LLPR}$: $(\forall x \in \mathbb{R})(x \geq 0 \mathbb{V} x \leq 0)$
$\updownarrow$	$\Longleftrightarrow$
NIL	$\mathbb{NIL}$
$(\forall x, y \in \mathbb{R})(xy = 0 \rightarrow x = 0 \vee y = 0)$	$(\forall x, y \in \mathbb{R})(xy = 0 \Rightarrow x = 0 \mathbb{V} y = 0)$
$\updownarrow$	$\Longleftrightarrow$
IVT: Intermediate value theorem (int. value computed by algo)	$\mathbb{IVT}$: Intermediate value theorem

Constructive Reverse Mathematics II

BISH (based on BHK)

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

(int. value computed by algo)

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LLPO}$

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

$\Longleftrightarrow$

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

$\Longleftrightarrow$

$\mathbb{NIL}$

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

$\Longleftrightarrow$

$\mathbb{IVT}$: Intermediate value theorem

(int. value computed by Ω-inv. proc.)

Constructive Reverse Mathematics II

BISH (based on BHK)

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

$\updownarrow$ (int. value computed by algo)

WKL

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LLPO}$

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

$\Longleftrightarrow$

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

$\Longleftrightarrow$

$\mathbb{NIL}$

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

$\Longleftrightarrow$

$\mathbb{IVT}$: Intermediate value theorem

(int. value computed by Ω-inv. proc.)

$\Longleftrightarrow$ $\mathbb{WKL}$

Constructive Reverse Mathematics II

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

Constructive Reverse Mathematics II

BISH (based on BHK)

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

$\updownarrow$ (int. value computed by algo)

WKL

$\mathbb{NSA}$ (based on CL)

Transfer Principle

$\mathbb{LLPO}$

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

$\Longleftrightarrow$

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

$\Longleftrightarrow$

$\mathbb{NIL}$

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

$\Longleftrightarrow$

$\mathbb{IVT}$: Intermediate value theorem

(int. value computed by Ω-inv. proc.)

$\Longleftrightarrow$ $\mathbb{WKL}$ $\Longleftrightarrow$ $\vee$-Transfer

BISH and $\mathbb{NSA}$ can prove $(\forall k \in \mathbb{N})(\exists x_0 \in [0,1])(|f(x_0)| < \frac{1}{k})$.

Constructive Reverse Mathematics III

Constructive Reverse Mathematics III

BISH (based on BHK)

non-constructive/non-algorithmic

$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics III

BISH (based on BHK)

non-constructive/non-algorithmic

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

$\updownarrow$

$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics III

BISH (based on BHK)

non-constructive/non-algorithmic

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

$\updownarrow$

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

$\updownarrow$

$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics III

BISH (based on BHK)

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

$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics III

BISH (based on BHK)

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

$\mathbb{NSA}$ (based on CL)

Transfer Principle

Constructive Reverse Mathematics III

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)
non-constructive/non-algorithmic	Transfer Principle
MP: For $P \in \Sigma_1$, $\neg\neg P \to P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rightarrow P$
$\updownarrow$	$\Longleftrightarrow$
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 BHK)

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

$\mathbb{NSA}$ (based on CL)

Transfer Principle

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

$\Longleftrightarrow$

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

$\Longleftrightarrow$

Constructive Reverse Mathematics III

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

Constructive Reverse Mathematics III

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)
non-constructive/non-algorithmic	Transfer Principle
MP: For $P \in \Sigma_1$, $\neg\neg P \rightarrow P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rightarrow P$
$\updownarrow$	$\Longleftrightarrow$
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \rightarrow x > 0)$	$\mathbb{MPR}$: $(\forall x \in \mathbb{R})(\sim\sim(x > 0) \Rightarrow x > 0)$
$\updownarrow$	$\Longleftrightarrow$
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 BHK)

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

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

$\updownarrow$

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

$\updownarrow$

$\mathbb{NSA}$ (based on CL)

Transfer Principle

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

$\Longleftrightarrow$

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

$\Longleftrightarrow$

$\mathbb{EXT}$: the extensionality theorem

Constructive Reverse Mathematics III

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

Constructive Reverse Mathematics III

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)
non-constructive/non-algorithmic	Transfer Principle
MP: For $P \in \Sigma_1$, $\neg\neg P \rightarrow P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rightarrow P$
$\updownarrow$	$\Longleftrightarrow$
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \rightarrow x > 0)$	$\mathbb{MPR}$: $(\forall x \in \mathbb{R})(\sim\sim(x > 0) \Rightarrow x > 0)$
$\updownarrow$	$\Longleftrightarrow$
EXT: the extensionality theorem	$\mathbb{EXT}$: the extensionality theorem
WLPO: For $P \in \Sigma_1$, $\neg\neg P \vee \neg P$	$\mathbb{WLPO}$: For $P \in \Sigma_1$, $\sim\sim P \,\mathbb{V} \sim P$
$\updownarrow$	$\Longleftrightarrow$
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.	

Constructive Reverse Mathematics III

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

Constructive Reverse Mathematics III

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)
non-constructive/non-algorithmic	Transfer Principle
MP: For $P \in \Sigma_1$, $\neg\neg P \to P$	$\mathbb{MP}$: For $P \in \Sigma_1$, $\sim\sim P \Rightarrow P$
$\updownarrow$	$\Longleftrightarrow$
MPR: $(\forall x \in \mathbb{R})(\neg\neg(x > 0) \to x > 0)$	$\mathbb{MPR}$: $(\forall x \in \mathbb{R})(\sim\sim(x > 0) \Rightarrow x > 0)$
$\updownarrow$	$\Longleftrightarrow$
EXT: the extensionality theorem	$\mathbb{EXT}$: the extensionality theorem
WLPO: For $P \in \Sigma_1$, $\neg\neg P \vee \neg P$	$\mathbb{WLPO}$: For $P \in \Sigma_1$, $\sim\sim P \,\mathbb{V} \sim P$
$\updownarrow$	$\Longleftrightarrow$
WLPR: $(\forall x \in \mathbb{R})\big[\neg\neg(x > 0) \vee \neg(x > 0)\big]$	$\mathbb{WLPR}$: $(\forall x \in \mathbb{R})\big[\sim\sim(x > 0) \,\mathbb{V} \sim(x > 0)\big]$
$\updownarrow$	$\Longleftrightarrow$
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 III

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

(Four Remarks)

Ω-invariance is weaker than Recursive

Ω-invariance is weaker than Recursive

Markov's principle MP can be reformulated as *If it is impossible that a TM runs forever, then it must halt*.

Ω-invariance is weaker than Recursive

Markov's principle MP can be reformulated as *If it is impossible that a TM runs forever, then it must halt*.

As no algorithmic upper bound on the halting time of the TM is given, MP is rejected in BISH.

Ω-invariance is weaker than Recursive

Markov's principle MP can be reformulated as *If it is impossible that a TM runs forever, then it must halt*.

As no algorithmic upper bound on the halting time of the TM is given, MP is rejected in BISH. The notion of algorithm in BISH is not identical to 'recursive'.

Ω-invariance is weaker than Recursive

Markov's principle MP can be reformulated as *If it is impossible that a TM runs forever, then it must halt*.

As no algorithmic upper bound on the halting time of the TM is given, MP is rejected in BISH. The notion of algorithm in BISH is not identical to 'recursive'.

Definition (In $\mathbb{NSA}$)

A formula ψ is $\mathbb{\Delta}_1$ if $\psi \iff (\exists n \in \mathbb{N})\varphi_1(n) \iff (\forall m \in \mathbb{N})\varphi_2(m)$.

Ω-invariance is weaker than Recursive

Markov's principle MP can be reformulated as *If it is impossible that a TM runs forever, then it must halt*.

As no algorithmic upper bound on the halting time of the TM is given, MP is rejected in BISH. The notion of algorithm in BISH is not identical to 'recursive'.

Definition (In $\mathbb{NSA}$)

A formula ψ is Δ_1 if $\psi \iff (\exists n \in \mathbb{N})\varphi_1(n) \iff (\forall m \in \mathbb{N})\varphi_2(m)$.

Theorem

In $\mathbb{NSA} + \mathbb{MP}$, all Δ_1-formulas are decidable.

Ω-invariance is weaker than Recursive

Markov's principle MP can be reformulated as *If it is impossible that a TM runs forever, then it must halt*.

As no algorithmic upper bound on the halting time of the TM is given, MP is rejected in BISH. The notion of algorithm in BISH is not identical to 'recursive'.

Definition (In $\mathbb{NSA}$)

A formula ψ is Δ_1 if $\psi \iff (\exists n \in \mathbb{N})\varphi_1(n) \iff (\forall m \in \mathbb{N})\varphi_2(m)$.

Theorem

In $\mathbb{NSA} + \mathbb{MP}$, all Δ_1-formulas are decidable.

But $\mathbb{MP}$ is not available in $\mathbb{NSA}$!

Ω-invariance is weaker than Recursive

Markov's principle MP can be reformulated as *If it is impossible that a TM runs forever, then it must halt*.

As no algorithmic upper bound on the halting time of the TM is given, MP is rejected in BISH. The notion of algorithm in BISH is not identical to 'recursive'.

Definition (In $\mathbb{NSA}$)

A formula ψ is Δ_1 if $\psi \iff (\exists n \in \mathbb{N})\varphi_1(n) \iff (\forall m \in \mathbb{N})\varphi_2(m)$.

Theorem

In $\mathbb{NSA} + \mathbb{MP}$, all Δ_1-formulas are decidable.

But $\mathbb{MP}$ is not available in $\mathbb{NSA}$!

Examples of non-Ω-invariant procedures?

Constructive Reverse Mathematics IV

Constructive Reverse Mathematics IV

Same for WMP, FAN_{Δ}, BD-N, and $\mathrm{MP}^{\vee}$.

Constructive Reverse Mathematics IV

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

Constructive Reverse Mathematics IV

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

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics IV

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

BISH (based on BHK)	$\mathbb{NSA}$ (based on CL)
LPO $\leftrightarrow$ MP+WLPO	

Constructive Reverse Mathematics IV

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

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

Constructive Reverse Mathematics IV

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

BISH (based on BHK)

LPO $\leftrightarrow$ MP+WLPO
MP $\leftrightarrow$ WMP + $\text{MP}^\vee$
WLPO $\rightarrow$ LLPO

$\mathbb{NSA}$ (based on CL)

Constructive Reverse Mathematics IV

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

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

Constructive Reverse Mathematics IV

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

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

Constructive Reverse Mathematics IV

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

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

Constructive Reverse Mathematics IV

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

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

Constructive Reverse Mathematics IV

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

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

Conclusion: $\mathbb{NSA} \approx$ BISH

Our interpretation from BISH to $\mathbb{NSA}$ has the following properties.

1. CRM-equivalences are preserved.

Conclusion: $\mathbb{NSA} \approx$ BISH

Our interpretation from BISH to $\mathbb{NSA}$ has the following properties.

1. CRM-equivalences are preserved.
2. Non-constructive princ. from BISH are interpreted as Transfer Princ. rejected in $\mathbb{NSA}$.

Conclusion: $\mathbb{NSA} \approx$ BISH

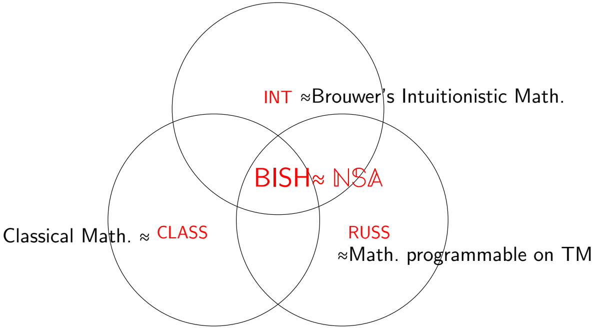
Our interpretation from BISH to $\mathbb{NSA}$ has the following properties.

1. CRM-equivalences are preserved.
2. Non-constructive princ. from BISH are interpreted as Transfer Princ. rejected in $\mathbb{NSA}$.
3. "Not all Δ_1-formulas are decidable" is preserved.

Conclusion: $\mathbb{NSA} \approx$ BISH

Our interpretation from BISH to $\mathbb{NSA}$ has the following properties.

1. CRM-equivalences are preserved.
2. Non-constructive princ. from BISH are interpreted as Transfer Princ. rejected in $\mathbb{NSA}$.
3. "Not all Δ_1-formulas are decidable" is preserved.



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.

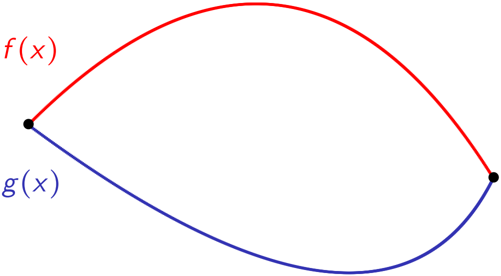
Can Ω-invariance help capture e.g. Type Theory?

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:

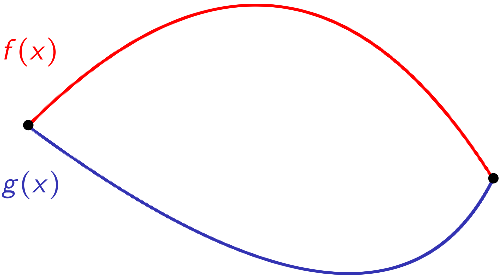


Future work: Type Theory

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 ($t \in [0,1]$).

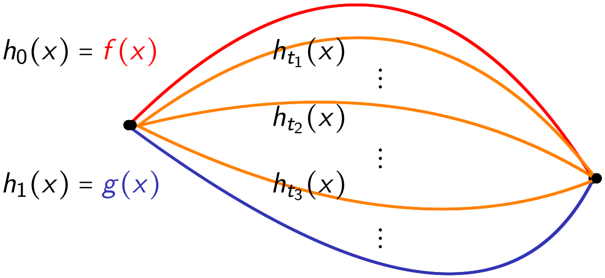


Future work: Type Theory

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 ($t \in [0,1]$).



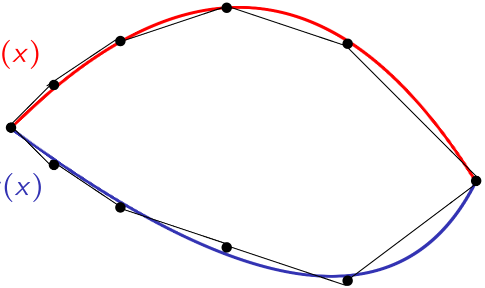
Future work: Type Theory

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

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

Homotopy:

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



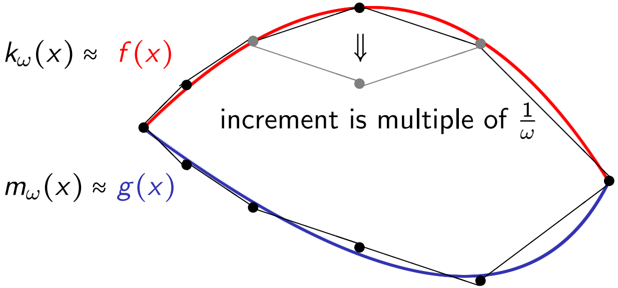
$m_\omega(x) \approx g(x)$

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:

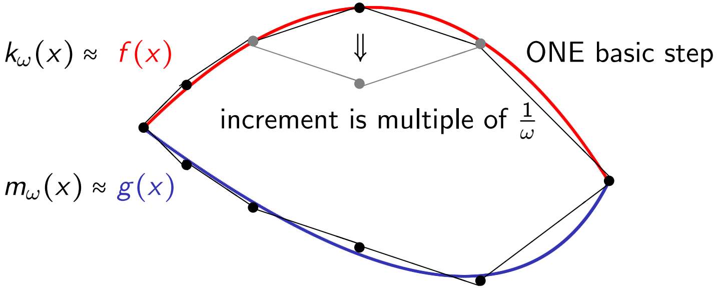


Future work: Type Theory

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

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

Homotopy:

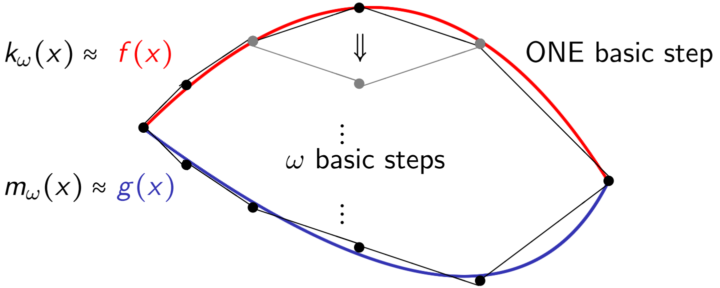


Future work: Type Theory

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

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

Homotopy:

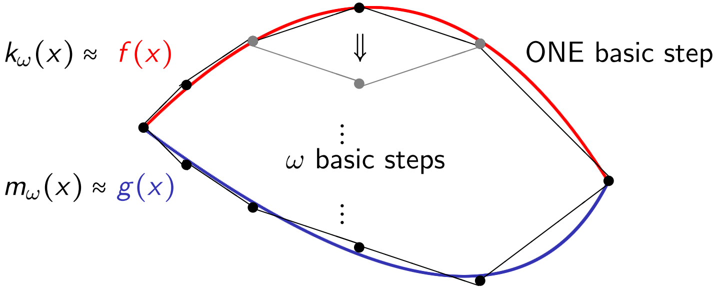


Future work: Type Theory

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

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

Homotopy:



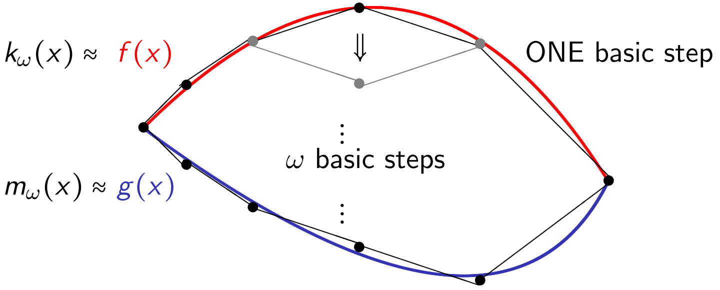
Independent of the choice of ω

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.



Independent of the choice of ω

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, most of Physics can be formalized in BISH (e.g. Gleason's thm).

Philosophy of Physics

Why is Mathematics in Physics so constructive/computable?

Indeed, most of Physics can be formalized in BISH (e.g. Gleason's thm).

Yet, in Physics, an informal version of NSA is used to date.

Philosophy of Physics

Why is Mathematics in Physics so constructive/computable?

Indeed, most of Physics can be formalized in BISH (e.g. Gleason's thm).

Yet, in Physics, an informal version of NSA is used to date.
(Weierstraß' notorious 'ε-δ' method was never adopted, neither was BISH).

Philosophy of Physics

Why is Mathematics in Physics so constructive/computable?

Indeed, most of Physics can be formalized in BISH (e.g. Gleason's thm).

Yet, in Physics, an informal version of NSA is used to date. (Weierstraß' notorious 'ε-δ' method was never adopted, neither was BISH).

Now, 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, most of Physics can be formalized in BISH (e.g. Gleason's thm).

Yet, in Physics, an informal version of NSA is used to date. (Weierstraß' notorious 'ε-δ' method was never adopted, neither was BISH).

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

A mathematical result with physical meaning will not depend on the choice of infinite number/infinitesimal used, i.e. it is Ω-invariant.

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

This research is generously sponsored by the John Templeton Foundation.

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

This research is generously sponsored by the John Templeton Foundation.

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

This research is generously sponsored by the John Templeton Foundation.

Thank you for your attention!

Any questions?