A new notion of reducibility for third-order arithmetic

Sam Sanders

RUB Bochum, Department of Philosophy II

CCA26,
Trier, July 2026

In a nutshell

This talk:

In a nutshell

This talk: a new formal notion capturing

Theorem A is computationally as strong as Theorem B,

In a nutshell

This talk: a new formal notion capturing

Theorem A is computationally as strong as Theorem B,

This notion is called N-reduction and combines the advantages of existing approaches going back to Turing and Kleene.

In a nutshell

This talk: a new formal notion capturing

Theorem A is computationally as strong as Theorem B,

This notion is called N-reduction and combines the advantages of existing approaches going back to Turing and Kleene. The goal is to generalise Weihrauch reducibility to third-order arithmetic.

In a nutshell

This talk: a new formal notion capturing

Theorem A is computationally as strong as Theorem B,

This notion is called N-reduction and combines the advantages of existing approaches going back to Turing and Kleene. The goal is to generalise Weihrauch reducibility to third-order arithmetic.

Comparison with known third-order frameworks due to Fujiwara and Kihara.

In a nutshell

This talk: a new formal notion capturing

Theorem A is computationally as strong as Theorem B,

This notion is called N-reduction and combines the advantages of existing approaches going back to Turing and Kleene. The goal is to generalise Weihrauch reducibility to third-order arithmetic.

Comparison with known third-order frameworks due to Fujiwara and Kihara.

We first consider Turing and Kleene computability and motivate the need for N-reduction along the way.

In a nutshell

This talk: a new formal notion capturing

Theorem A is computationally as strong as Theorem B,

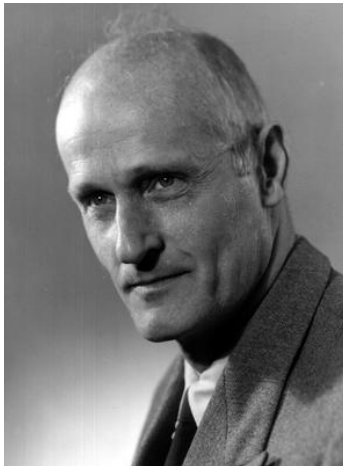
This notion is called N-reduction and combines the advantages of existing approaches going back to Turing and Kleene. The goal is to generalise Weihrauch reducibility to third-order arithmetic.

Comparison with known third-order frameworks due to Fujiwara and Kihara.

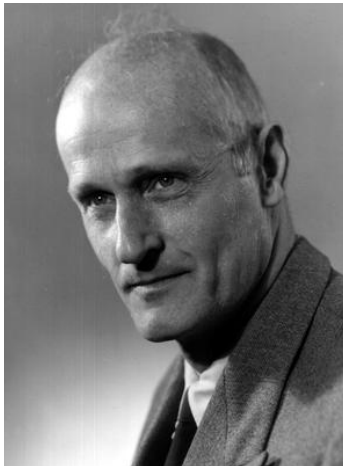
We first consider Turing and Kleene computability and motivate the need for N-reduction along the way.

A lot of these results are based on jww Dag Normann (Oslo).

Turing and Kleene

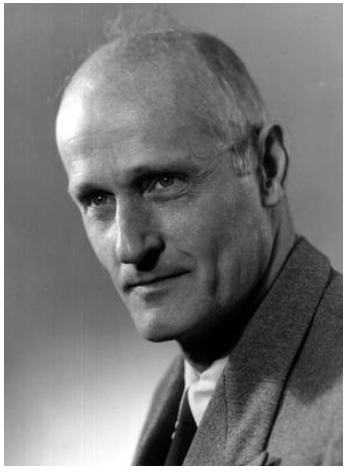


Turing and Kleene



Turing's 'machine' framework: first intuitively convincing notion of computing with real numbers (Entscheidungsproblem).

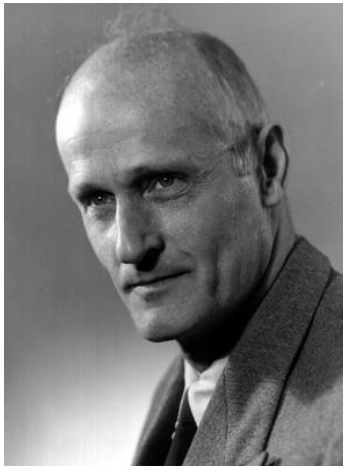
Turing and Kleene



Turing's 'machine' framework: first intuitively convincing notion of computing with real numbers (Entscheidungsproblem).

Kleene S1-S9: extends to computing with objects of all finite types.

Turing and Kleene

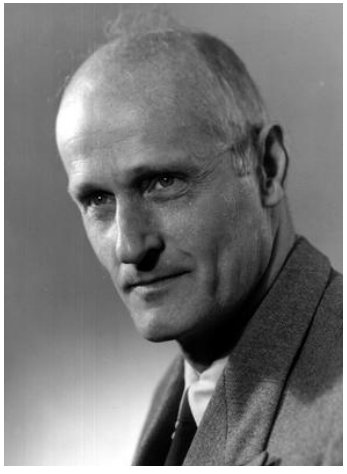


Turing's 'machine' framework: first intuitively convincing notion of computing with real numbers (Entscheidungsproblem).

Kleene S1-S9: extends to computing with objects of all finite types.

Infinite time Turing machine (ITTM): running time measured by ordinals (Hamkins, Lewis);

Turing and Kleene



Turing's 'machine' framework: first intuitively convincing notion of computing with real numbers (Entscheidungsproblem).

Kleene S1-S9: extends to computing with objects of all finite types.

Infinite time Turing machine (ITTM): running time measured by ordinals (Hamkins, Lewis); higher-order inputs possible (Welsh)

Problems with existing frameworks

Problems with existing frameworks

ITTMs are too strong:

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems:

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems: no T-predicate

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems: no T-predicate, hard-coding of the recursion theorem via S9 (Normann),

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems: no T-predicate, hard-coding of the recursion theorem via S9 (Normann), and no (canonical) complexity theory.

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems: no T-predicate, hard-coding of the recursion theorem via S9 (Normann), and no (canonical) complexity theory.

TMs rely too heavily on coding:

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems: no T-predicate, hard-coding of the recursion theorem via S9 (Normann), and no (canonical) complexity theory.

TMs rely too heavily on coding: $\mathbb{R} \to \mathbb{R}$-functions have to be represented as reals, greatly changing the computational power/strength/hardness of the associated theorems.

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems: no T-predicate, hard-coding of the recursion theorem via S9 (Normann), and no (canonical) complexity theory.

TMs rely too heavily on coding: $\mathbb{R} \to \mathbb{R}$-functions have to be represented as reals, greatly changing the computational power/strength/hardness of the associated theorems.

The first and second point are due to Normann; the third point is due to Normann-Sanders.

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems: no T-predicate, hard-coding of the recursion theorem via S9 (Normann), and no (canonical) complexity theory.

TMs rely too heavily on coding: $\mathbb{R} \to \mathbb{R}$-functions have to be represented as reals, greatly changing the computational power/strength/hardness of the associated theorems.

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems: no T-predicate, hard-coding of the recursion theorem via S9 (Normann), and no (canonical) complexity theory.

TMs rely too heavily on coding: $\mathbb{R} \to \mathbb{R}$-functions have to be represented as reals, greatly changing the computational power/strength/hardness of the associated theorems.

The best of Turing and Kleene:

- Computing with actual $\mathbb{R} \to \mathbb{R}$ (and $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$) functions, NOT codes.

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems: no T-predicate, hard-coding of the recursion theorem via S9 (Normann), and no (canonical) complexity theory.

TMs rely too heavily on coding: $\mathbb{R} \to \mathbb{R}$-functions have to be represented as reals, greatly changing the computational power/strength/hardness of the associated theorems.

The best of Turing and Kleene:

- Computing with actual $\mathbb{R} \to \mathbb{R}$ (and $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$) functions, NOT codes.
- Computation that is close to TMs, thus avoiding the above conceptual problems (caused by S9).

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems: no T-predicate, hard-coding of the recursion theorem via S9 (Normann), and no (canonical) complexity theory.

TMs rely too heavily on coding: $\mathbb{R} \to \mathbb{R}$-functions have to be represented as reals, greatly changing the computational power/strength/hardness of the associated theorems.

The best of Turing and Kleene:

- Computing with actual $\mathbb{R} \to \mathbb{R}$ (and $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$) functions, NOT codes.
- Computation that is close to TMs, thus avoiding the above conceptual problems (caused by S9).
- Avoid fourth-order objects if possible (Friedman et al).

Problems with existing frameworks

ITTMs are too strong: they outright compute (too) many basic operations from real analysis that are not S1-S9 computable.

Kleene's S1-S9 has conceptual problems: no T-predicate, hard-coding of the recursion theorem via S9 (Normann), and no (canonical) complexity theory.

TMs rely too heavily on coding: $\mathbb{R} \to \mathbb{R}$-functions have to be represented as reals, greatly changing the computational power/strength/hardness of the associated theorems.

The best of Turing and Kleene:

- Computing with actual $\mathbb{R} \to \mathbb{R}$ (and $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$) functions, NOT codes.
- Computation that is close to TMs, thus avoiding the above conceptual problems (caused by S9).
- Avoid fourth-order objects if possible (Friedman et al).

In a nutshell

In a nutshell

This talk: a new formal notion capturing

Theorem A is computationally as strong as Theorem B,

This notion is called N-reduction and combines the advantages of existing approaches going back to Turing and Kleene.

In a nutshell

This talk: a new formal notion capturing

Theorem A is computationally as strong as Theorem B,

This notion is called N-reduction and combines the advantages of existing approaches going back to Turing and Kleene.

The goal is to generalise Weihrauch reducibility to third-order arithmetic.

Higher types redux

Higher types redux

Many third-order thms of real analysis can be formulated as:

$$(\forall Y : \mathbb{N}^{\mathbb{N}} \to \mathbb{N})(\exists x \in \mathbb{N}^{\mathbb{N}})A(Y, x),$$

ignoring representations of reals etc.

Higher types redux

Many third-order thms of real analysis can be formulated as:

$$(\forall Y : \mathbb{N}^{\mathbb{N}} \to \mathbb{N})(\exists x \in \mathbb{N}^{\mathbb{N}})A(Y, x),$$

ignoring representations of reals etc.

Higher-order RM then studies whether we have:

$$(\forall Y^2)(\exists x^1)A(Y, x) \to (\forall Z^2)(\exists y^1)B(Z, y),$$

Higher types redux

Many third-order thms of real analysis can be formulated as:

$$(\forall Y : \mathbb{N}^{\mathbb{N}} \to \mathbb{N})(\exists x \in \mathbb{N}^{\mathbb{N}})A(Y, x),$$

ignoring representations of reals etc.

Higher-order RM then studies whether we have:

$$(\forall Y^2)(\exists x^1)A(Y, x) \to (\forall Z^2)(\exists y^1)B(Z, y),$$

while higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y, \Theta(Y)) \to (\forall Z^2)B(Z, \{e\}(\Theta, Z))\big] \qquad \text{(K)}$$

where $\{e\}(\Theta, Z)$ is the e-th Kleene S1-S9 algorithm.

Higher types redux

Many third-order thms of real analysis can be formulated as:

$$(\forall Y : \mathbb{N}^{\mathbb{N}} \to \mathbb{N})(\exists x \in \mathbb{N}^{\mathbb{N}})A(Y, x),$$

ignoring representations of reals etc.

Higher-order RM then studies whether we have:

$$(\forall Y^2)(\exists x^1)A(Y, x) \to (\forall Z^2)(\exists y^1)B(Z, y),$$

while higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y, \Theta(Y)) \to (\forall Z^2)B(Z, \{e\}(\Theta, Z))\big] \qquad \text{(K)}$$

where $\{e\}(\Theta, Z)$ is the e-th Kleene S1-S9 algorithm.

We want a version of (K) involving Turing machines.

Higher types redux-ish

Higher-order RM studies whether we have:

$$(\forall Y^2)(\exists x^1)A(Y,x) \to (\forall Z^2)(\exists y^1)B(Z,x),$$

while higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y,\Theta(Y)) \to (\forall Z^2)B(Z,\{e\}(\Theta,Z))\big] \qquad (\mathsf{K})$$

where $\{e\}(\Theta,Z)$ is a Kleene S1-S9 algorithm.

Higher types redux-ish

Higher-order RM studies whether we have:

$$(\forall Y^2)(\exists x^1)A(Y,x) \to (\forall Z^2)(\exists y^1)B(Z,x),$$

while higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y,\Theta(Y)) \to (\forall Z^2)B(Z,\{e\}(\Theta,Z))\big] \qquad \text{(K)}$$

where $\{e\}(\Theta,Z)$ is a Kleene S1-S9 algorithm.

A version of (K) involving Turing machines:

$$(\forall Z^2,x^1)\big[A(t(Z),x) \to [\{e_0\}^{s(x,Z)}\downarrow \wedge\, B(Z,\{e_0\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where $t^{2\to 2}, s^{(1\times 2)\to 1}$ are primitive recursive and $\{e_0\}^{s(x,Z)}$ is a Turing machine with oracle $s(x,Z)$.

Higher types redux-ish

Higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y,\Theta(Y)) \to (\forall Z^2)B(Z,\{e\}(\Theta,Z))\big] \qquad \text{(K)}$$

where $\{e\}(\Theta, Z)$ is a Kleene S1-S9 algorithm.

A special case of (K) involving Turing machines:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e_0\}^{s(x,Z)}\downarrow \wedge B(Z,\{e_0\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e_0\}^y$ is a Turing machine with oracle y^1.

Higher types redux-ish

Higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y,\Theta(Y)) \to (\forall Z^2)B(Z,\{e\}(\Theta,Z))\big] \qquad \text{(K)}$$

where $\{e\}(\Theta, Z)$ is a Kleene S1-S9 algorithm.

A special case of (K) involving Turing machines:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e_0\}^{s(x,Z)}\downarrow \wedge B(Z,\{e_0\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e_0\}^y$ is a Turing machine with oracle y^1.

Intuitive meaning of (N) for fixed Z:

Higher types redux-ish

Higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y,\Theta(Y)) \to (\forall Z^2)B(Z,\{e\}(\Theta,Z))\big] \qquad \text{(K)}$$

where $\{e\}(\Theta, Z)$ is a Kleene S1-S9 algorithm.

A special case of (K) involving Turing machines:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e_0\}^{s(x,Z)}\downarrow \wedge B(Z,\{e_0\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e_0\}^y$ is a Turing machine with oracle y^1.

Intuitive meaning of (N) for fixed Z:

- Antecedent of (K) is too strong: we only need $(\forall Y)A(Y,\Theta(Y))$ for one instance $t(Z)$ and associated x.

Higher types redux-ish

Higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y,\Theta(Y)) \to (\forall Z^2)B(Z,\{e\}(\Theta,Z))\big] \qquad \text{(K)}$$

where $\{e\}(\Theta, Z)$ is a Kleene S1-S9 algorithm.

A special case of (K) involving Turing machines:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e_0\}^{s(x,Z)}\downarrow \wedge B(Z,\{e_0\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e_0\}^y$ is a Turing machine with oracle y^1.

Intuitive meaning of (N) for fixed Z:

- Antecedent of (K) is too strong: we only need $(\forall Y)A(Y,\Theta(Y))$ for one instance $t(Z)$ and associated x.
- Consequent of (K) then reduces to Turing computability.

Higher types redux-ish

Higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y,\Theta(Y)) \to (\forall Z^2)B(Z,\{e\}(\Theta,Z))\big] \qquad \text{(K)}$$

where $\{e\}(\Theta, Z)$ is a Kleene S1-S9 algorithm.

A version of (K) involving Turing machines:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e_0\}^{s(x,Z)}\downarrow \wedge\, B(Z,\{e_0\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e_0\}^y$ is a Turing machine with oracle y^1.

Higher types redux-ish

Higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y,\Theta(Y)) \to (\forall Z^2)B(Z,\{e\}(\Theta,Z))\big] \qquad \text{(K)}$$

where $\{e\}(\Theta, Z)$ is a Kleene S1-S9 algorithm.

A version of (K) involving Turing machines:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e_0\}^{s(x,Z)}\downarrow \wedge\, B(Z,\{e_0\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e_0\}^y$ is a Turing machine with oracle y^1.

In case (N), we say that

solving the problem $(\forall Z^2)(\exists y^1)B(Z,x)$ *N*-reduces to solving the problem $(\forall Y^2)(\exists x^1)A(Y,x)$.

The scope of N-reduction

Solving the problem $(\forall Z^2)(\exists y^1)B(Z,x)$ N-reduces to solving the problem $(\forall Y^2)(\exists x^1)A(Y,x)$.

in case:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e\}^{s(x,Z)}\downarrow \wedge\, B(Z,\{e\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e\}^y$ is a TM with oracle y^1.

The scope of N-reduction

Solving the problem $(\forall Z^2)(\exists y^1)B(Z,x)$ N-reduces to solving the problem $(\forall Y^2)(\exists x^1)A(Y,x)$.

in case:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e\}^{s(x,Z)}\downarrow \wedge\, B(Z,\{e\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e\}^y$ is a TM with oracle y^1.

How general is (N)?

The scope of N-reduction

Solving the problem $(\forall Z^2)(\exists y^1)B(Z,x)$ N-reduces to solving the problem $(\forall Y^2)(\exists x^1)A(Y,x)$.

in case:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e\}^{s(x,Z)}\downarrow \wedge\, B(Z,\{e\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e\}^y$ is a TM with oracle y^1.

How general is (N)?

- With some effort, many S1-S9 and RM results 'carry over' to N-reduction.

The scope of N-reduction

Solving the problem $(\forall Z^2)(\exists y^1)B(Z,x)$ N-reduces to solving the problem $(\forall Y^2)(\exists x^1)A(Y,x)$.

in case:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e\}^{s(x,Z)}\downarrow \wedge B(Z,\{e\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e\}^y$ is a TM with oracle y^1.

How general is (N)?

- With some effort, many S1-S9 and RM results 'carry over' to N-reduction.
- Grilliot's trick regarding discontinuous functions yields N-reductions for Brouwer's intutionistic mathematics.

The scope of N-reduction

Solving the problem $(\forall Z^2)(\exists y^1)B(Z,x)$ N-reduces to solving the problem $(\forall Y^2)(\exists x^1)A(Y,x)$.

in case:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e\}^{s(x,Z)}\downarrow \wedge B(Z,\{e\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e\}^y$ is a TM with oracle y^1.

How general is (N)?

- With some effort, many S1-S9 and RM results 'carry over' to N-reduction.
- Grilliot's trick regarding discontinuous functions yields N-reductions for Brouwer's intutionistic mathematics.
- Sometimes: 'strong N-reduction' where $s(x,Z) = u(x)$.

The scope of N-reduction

Solving the problem $(\forall Z^2)(\exists y^1)B(Z,x)$ N-reduces to solving the problem $(\forall Y^2)(\exists x^1)A(Y,x)$.

in case:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e\}^{s(x,Z)}\downarrow \wedge B(Z,\{e\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e\}^y$ is a TM with oracle y^1.

How general is (N)?

- With some effort, many S1-S9 and RM results 'carry over' to N-reduction.
- Grilliot's trick regarding discontinuous functions yields N-reductions for Brouwer's intutionistic mathematics.
- Sometimes: 'strong N-reduction' where $s(x,Z) = u(x)$.
- Sometimes: 'μN-reduction' where we allow $t(Z, \exists^2)$.

Classical

Suprema, jumps, and comprehension

The following (strongly) μN-reduce to each other:

Classical

Suprema, jumps, and comprehension

The following (strongly) μN-reduce to each other:

- For Y^2, the set $\{n \in \mathbb{N} : (\exists x \in \mathbb{N}^{\mathbb{N}})(Y(x, n) = 0)\}$ exists.

Classical

Suprema, jumps, and comprehension

The following (strongly) μN-reduce to each other:

- For Y^2, the set $\{n \in \mathbb{N} : (\exists x \in \mathbb{N}^{\mathbb{N}})(Y(x, n) = 0)\}$ exists.
- (Darboux, 1872) For $f : [0, 1] \to [0, 1]$, $\sup_{x \in [0,1]} f(x)$ exists.

Classical

Suprema, jumps, and comprehension

The following (strongly) μN-reduce to each other:

- For Y^2, the set $\{n \in \mathbb{N} : (\exists x \in \mathbb{N}^{\mathbb{N}})(Y(x, n) = 0)\}$ exists.
- (Darboux, 1872) For $f : [0, 1] \to [0, 1]$, $\sup_{x \in [0,1]} f(x)$ exists.
- Convergence theorems for nets indexed by $\mathbb{N}^{\mathbb{N}}$ (Moore-Smith).

Classical

Suprema, jumps, and comprehension

The following (strongly) μN-reduce to each other:

- For Y^2, the set $\{n \in \mathbb{N} : (\exists x \in \mathbb{N}^{\mathbb{N}})(Y(x, n) = 0)\}$ exists.
- (Darboux, 1872) For $f : [0, 1] \to [0, 1]$, $\sup_{x \in [0,1]} f(x)$ exists.
- Convergence theorems for nets indexed by $\mathbb{N}^{\mathbb{N}}$ (Moore-Smith).
- ...

The following (strongly) μN-reduce to each other:

Suprema, jumps, and comprehension

The following (strongly) μN-reduce to each other:

- For Y^2, the set $\{n \in \mathbb{N} : (\exists x \in \mathbb{N}^{\mathbb{N}})(Y(x, n) = 0)\}$ exists.
- (Darboux, 1872) For $f : [0, 1] \to [0, 1]$, $\sup_{x \in [0,1]} f(x)$ exists.
- Convergence theorems for nets indexed by $\mathbb{N}^{\mathbb{N}}$ (Moore-Smith).
- ...

The following (strongly) μN-reduce to each other:

- A countable (=injection to $\mathbb{N}$) subset of $[0, 1]$ has an enumeration.

Classical

Suprema, jumps, and comprehension

The following (strongly) μN-reduce to each other:

- For Y^2, the set $\{n \in \mathbb{N} : (\exists x \in \mathbb{N}^{\mathbb{N}})(Y(x, n) = 0)\}$ exists.
- (Darboux, 1872) For $f : [0, 1] \to [0, 1]$, $\sup_{x \in [0,1]} f(x)$ exists.
- Convergence theorems for nets indexed by $\mathbb{N}^{\mathbb{N}}$ (Moore-Smith).
- ...

The following (strongly) μN-reduce to each other:

- A countable (=injection to $\mathbb{N}$) subset of $[0, 1]$ has an enumeration.
- For $f \in BV$, $\sup_{x \in [0,1]} f(x)$ exists.

Suprema, jumps, and comprehension

The following (strongly) μN-reduce to each other:

- For Y^2, the set $\{n \in \mathbb{N} : (\exists x \in \mathbb{N}^{\mathbb{N}})(Y(x, n) = 0)\}$ exists.
- (Darboux, 1872) For $f : [0, 1] \to [0, 1]$, $\sup_{x \in [0,1]} f(x)$ exists.
- Convergence theorems for nets indexed by $\mathbb{N}^{\mathbb{N}}$ (Moore-Smith).
- ...

The following (strongly) μN-reduce to each other:

- A countable (=injection to $\mathbb{N}$) subset of $[0, 1]$ has an enumeration.
- For $f \in BV$, $\sup_{x \in [0,1]} f(x)$ exists.
- Convergence theorems for nets indexed by countable sets.

Suprema, jumps, and comprehension

The following (strongly) μN-reduce to each other:

- For Y^2, the set $\{n \in \mathbb{N} : (\exists x \in \mathbb{N}^{\mathbb{N}})(Y(x, n) = 0)\}$ exists.
- (Darboux, 1872) For $f : [0, 1] \to [0, 1]$, $\sup_{x \in [0,1]} f(x)$ exists.
- Convergence theorems for nets indexed by $\mathbb{N}^{\mathbb{N}}$ (Moore-Smith).
- ...

The following (strongly) μN-reduce to each other:

- A countable (=injection to $\mathbb{N}$) subset of $[0, 1]$ has an enumeration.
- For $f \in BV$, $\sup_{x \in [0,1]} f(x)$ exists.
- Convergence theorems for nets indexed by countable sets.
- ...

Functions of bounded variation (BV) were introduced in 1881 by Jordan when studying the convergence of Fourier series.

Non-classical

Grilliot's trick

Grilliot's trick

Grilliot's trick = a method for obtaining Turing jump functional E (aka Kleene's $\exists^2$) from a discontinuous function on $\mathbb{R}$ or $\mathbb{N}^{\mathbb{N}}$.

$$(\forall f : \mathbb{N} \to \mathbb{N})(E(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)). \qquad (\exists^2)$$

Grilliot's trick

Grilliot's trick = a method for obtaining Turing jump functional E (aka Kleene's $\exists^2$) from a discontinuous function on $\mathbb{R}$ or $\mathbb{N}^{\mathbb{N}}$.

$$(\forall f : \mathbb{N} \to \mathbb{N})(E(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)). \qquad (\exists^2)$$

Kohlenbach's observation: Grilliot's trick is highly effective and can be formalised in weak systems.

Grilliot's trick

Grilliot's trick = a method for obtaining Turing jump functional E (aka Kleene's $\exists^2$) from a discontinuous function on $\mathbb{R}$ or $\mathbb{N}^{\mathbb{N}}$.

$$(\forall f : \mathbb{N} \to \mathbb{N})(E(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)). \qquad (\exists^2)$$

Kohlenbach's observation: Grilliot's trick is highly effective and can be formalised in weak systems.

Kohlenbach's work yields N-reductions between e.g.

- All $\mathbb{R} \to \mathbb{R}$-functions are continuous (aka $\neg(\exists^2)$, Brouwer).

Grilliot's trick

Grilliot's trick = a method for obtaining Turing jump functional E (aka Kleene's $\exists^2$) from a discontinuous function on $\mathbb{R}$ or $\mathbb{N}^{\mathbb{N}}$.

$$(\forall f : \mathbb{N} \to \mathbb{N})(E(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)). \qquad (\exists^2)$$

Kohlenbach's observation: Grilliot's trick is highly effective and can be formalised in weak systems.

Kohlenbach's work yields N-reductions between e.g.

- All $\mathbb{R} \to \mathbb{R}$-functions are continuous (aka $\neg(\exists^2)$, Brouwer).
- The negation of the uniform intermediate value theorem.

Grilliot's trick

Grilliot's trick = a method for obtaining Turing jump functional E (aka Kleene's $\exists^2$) from a discontinuous function on $\mathbb{R}$ or $\mathbb{N}^{\mathbb{N}}$.

$$(\forall f : \mathbb{N} \to \mathbb{N})(E(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)). \qquad (\exists^2)$$

Kohlenbach's observation: Grilliot's trick is highly effective and can be formalised in weak systems.

Kohlenbach's work yields N-reductions between e.g.

- All $\mathbb{R} \to \mathbb{R}$-functions are continuous (aka $\neg(\exists^2)$, Brouwer).
- The negation of the uniform intermediate value theorem.
- The negation of the uniform Weierstrass maximum theorem.
- ...

Grilliot's trick

Grilliot's trick = a method for obtaining Turing jump functional E (aka Kleene's $\exists^2$) from a discontinuous function on $\mathbb{R}$ or $\mathbb{N}^{\mathbb{N}}$.

$$(\forall f : \mathbb{N} \to \mathbb{N})(E(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)). \qquad (\exists^2)$$

Kohlenbach's observation: Grilliot's trick is highly effective and can be formalised in weak systems.

Kohlenbach's work yields N-reductions between e.g.

- All $\mathbb{R} \to \mathbb{R}$-functions are continuous (aka $\neg(\exists^2)$, Brouwer).
- The negation of the uniform intermediate value theorem.
- The negation of the uniform Weierstrass maximum theorem.
- ...

There are many equivalences for $(\exists^2)$ in HORM (see ...).

Conjectures

Our hope and dream: Turing computability results (help) settle 'too hard' questions, like the following:

Conjectures

Our hope and dream: Turing computability results (help) settle 'too hard' questions, like the following:

ADS^3: for an infinite linear ordering that is countable (=injection to $\mathbb{N}$), there is a (strictly) descending or increasing sequence.

Conjectures

Our hope and dream: Turing computability results (help) settle 'too hard' questions, like the following:

ADS^3: for an infinite linear ordering that is countable (=injection to $\mathbb{N}$), there is a (strictly) descending or increasing sequence.

NIN: there is no injection from $[0, 1]$ to $\mathbb{N}$

Conjectures

Our hope and dream: Turing computability results (help) settle 'too hard' questions, like the following:

ADS^3: for an infinite linear ordering that is countable (=injection to $\mathbb{N}$), there is a (strictly) descending or increasing sequence.

NIN: there is no injection from $[0,1]$ to $\mathbb{N}$

Cantor's theorem: for any countable $A \subset [0,1]$ (=injection to $\mathbb{N}$), there is $x \in \big([0,1] \setminus A\big)$

Conjectures

Our hope and dream: Turing computability results (help) settle 'too hard' questions, like the following:

ADS^3: for an infinite linear ordering that is countable (=injection to $\mathbb{N}$), there is a (strictly) descending or increasing sequence.

NIN: there is no injection from $[0,1]$ to $\mathbb{N}$

Cantor's theorem: for any countable $A \subset [0,1]$ (=injection to $\mathbb{N}$), there is $x \in \big([0,1] \setminus A\big)$

Despite the last two being equivalent in RM, we conjecture that

- NIN does not N-reduce to ADS^3 (and vice versa).

Conjectures

Our hope and dream: Turing computability results (help) settle 'too hard' questions, like the following:

ADS^3: for an infinite linear ordering that is countable (=injection to $\mathbb{N}$), there is a (strictly) descending or increasing sequence.

NIN: there is no injection from $[0,1]$ to $\mathbb{N}$

Cantor's theorem: for any countable $A \subset [0,1]$ (=injection to $\mathbb{N}$), there is $x \in \big([0,1] \setminus A\big)$

Despite the last two being equivalent in RM, we conjecture that

- NIN does not N-reduce to ADS^3 (and vice versa).
- NIN does not N-reduce to Cantor's theorem (and vice versa).

Conjectures

Our hope and dream: Turing computability results (help) settle 'too hard' questions, like the following:

ADS^3: for an infinite linear ordering that is countable (=injection to $\mathbb{N}$), there is a (strictly) descending or increasing sequence.

NIN: there is no injection from $[0,1]$ to $\mathbb{N}$

Cantor's theorem: for any countable $A \subset [0,1]$ (=injection to $\mathbb{N}$), there is $x \in \big([0,1] \setminus A\big)$

Despite the last two being equivalent in RM, we conjecture that

- NIN does not N-reduce to ADS^3 (and vice versa).
- NIN does not N-reduce to Cantor's theorem (and vice versa).

We have no idea how to tackle the S1-S9 versions of these conjectures.

Recall N-reduction

Solving the problem $(\forall Z^2)(\exists y^1)B(Z,x)$ N-reduces to solving the problem $(\forall Y^2)(\exists x^1)A(Y,x)$.

in case:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e\}^{s(x,Z)}\downarrow \wedge\, B(Z,\{e\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e\}^y$ is a TM with oracle y^1.

More conjectures

Consider the Heine-Borel theorem/Cousin's lemma (HBU):

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the 'canonical' covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \ldots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

More conjectures

Consider the Heine-Borel theorem/Cousin's lemma (HBU):

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the 'canonical' covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \dots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

NIN: there is no injection from $[0,1]$ to $\mathbb{N}$

More conjectures

Consider the Heine-Borel theorem/Cousin's lemma (HBU):

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the 'canonical' covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \dots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

NIN: there is no injection from $[0,1]$ to $\mathbb{N}$

Cantor's theorem: for any countable $A \subset [0,1]$ (=injection to $\mathbb{N}$), there is $x \in \big([0,1] \setminus A\big)$

More conjectures

Consider the Heine-Borel theorem/Cousin's lemma (HBU):

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the 'canonical' covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \dots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

NIN: there is no injection from $[0,1]$ to $\mathbb{N}$

Cantor's theorem: for any countable $A \subset [0,1]$ (=injection to $\mathbb{N}$), there is $x \in \big([0,1] \setminus A\big)$

We have the following:

- The problem NIN N-reduces to HBU.

More conjectures

Consider the Heine-Borel theorem/Cousin's lemma (HBU):

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the 'canonical' covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \dots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

NIN: there is no injection from $[0,1]$ to $\mathbb{N}$

Cantor's theorem: for any countable $A \subset [0,1]$ (=injection to $\mathbb{N}$), there is $x \in ([0,1] \setminus A)$

We have the following:

- The problem NIN N-reduces to HBU.
- Cantor's theorem N-reduces to HBU for Baire class 2.

More conjectures

Consider the Heine-Borel theorem/Cousin's lemma (HBU):

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the 'canonical' covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \dots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

NIN: there is no injection from $[0,1]$ to $\mathbb{N}$

Cantor's theorem: for any countable $A \subset [0,1]$ (=injection to $\mathbb{N}$), there is $x \in \big([0,1] \setminus A\big)$

We have the following:

- The problem NIN N-reduces to HBU.
- Cantor's theorem N-reduces to HBU for Baire class 2.

Conjecture: NIN does not N-reduce to HBU for Baire class 2.

More and more conjectures

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ *and the canonical covering* $\cup_{x\in[0,1]} B(x, \Psi(x))$, *there are* $x_0, \ldots, x_k \in [0,1]$ *such that* $\cup_{i\leq k} B(x_i, \Psi(x_i))$ *still covers* $[0,1]$, *i.e. finite sub-covering.*

More and more conjectures

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \ldots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

Consider the closely related Lebesgue number lemma (LNL):

Principle (LNL)

For $\Psi : [0,1] \to \mathbb{R}^+$, the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$ has a Lebesgue number, i.e. $\delta > 0$ such that every subset with diameter less than δ is contained in some member of the cover.

More and more conjectures

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \ldots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

Consider the closely related Lebesgue number lemma (LNL):

Principle (LNL)

For $\Psi : [0,1] \to \mathbb{R}^+$, the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$ has a Lebesgue number, i.e. $\delta > 0$ such that every subset with diameter less than δ is contained in some member of the cover.

From a finite sub-covering, one readily computes a Lebesgue number (via the minimal overlap in the former).

More and more conjectures

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \ldots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

Consider the closely related Lebesgue number lemma (LNL):

Principle (LNL)

For $\Psi : [0,1] \to \mathbb{R}^+$, the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$ has a Lebesgue number, i.e. $\delta > 0$ such that every subset with diameter less than δ is contained in some member of the cover.

From a finite sub-covering, one readily computes a Lebesgue number (via the minimal overlap in the former).

Hence, the Lebesgue number lemma LNL (strongly?) N-reduces to HBU

More and more conjectures

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \ldots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

Consider the closely related Lebesgue number lemma (LNL):

Principle (LNL)

For $\Psi : [0,1] \to \mathbb{R}^+$, the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$ has a Lebesgue number, i.e. $\delta > 0$ such that every subset with diameter less than δ is contained in some member of the cover.

From a finite sub-covering, one readily computes a Lebesgue number (via the minimal overlap in the former).

Hence, the Lebesgue number lemma LNL (strongly?) N-reduces to HBU (sensitive to the definition of covering).

More and more conjectures

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \ldots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

Principle (LNL)

For $\Psi : [0,1] \to \mathbb{R}^+$, the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$ has a Lebesgue number, i.e. $\delta > 0$ such that every subset with diameter less than δ is contained in some member of the cover.

More and more conjectures

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \ldots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

Principle (LNL)

For $\Psi : [0,1] \to \mathbb{R}^+$, the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$ has a Lebesgue number, i.e. $\delta > 0$ such that every subset with diameter less than δ is contained in some member of the cover.

- LNL strongly μN-reduces to monotone convergence theorem: a monotone net in $[0,1]$ indexed by $\mathbb{N}^{\mathbb{N}}$, has a limit.

More and more conjectures

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \ldots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

Principle (LNL)

For $\Psi : [0,1] \to \mathbb{R}^+$, the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$ has a Lebesgue number, i.e. $\delta > 0$ such that every subset with diameter less than δ is contained in some member of the cover.

- LNL strongly μN-reduces to monotone convergence theorem: a monotone net in $[0,1]$ indexed by $\mathbb{N}^{\mathbb{N}}$, has a limit.
- Conjecture: HBU does not N-reduce to this convergence theorem.

More and more conjectures

Principle (HBU)

For $\Psi : [0,1] \to \mathbb{R}^+$ and the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$, there are $x_0, \ldots, x_k \in [0,1]$ such that $\cup_{i\leq k} B(x_i, \Psi(x_i))$ still covers $[0,1]$, i.e. finite sub-covering.

Principle (LNL)

For $\Psi : [0,1] \to \mathbb{R}^+$, the canonical covering $\cup_{x\in[0,1]} B(x, \Psi(x))$ has a Lebesgue number, i.e. $\delta > 0$ such that every subset with diameter less than δ is contained in some member of the cover.

- LNL strongly μN-reduces to monotone convergence theorem: a monotone net in $[0,1]$ indexed by $\mathbb{N}^{\mathbb{N}}$, has a limit.
- Conjecture: HBU does not N-reduce to this convergence theorem.
- Conjecture: HBU does not N-reduce to LNL.

Origins of N-reduction

Origins of N-reduction

Kreisel-Troelstra introduce the neighbourhood function principle NFP as a more constructive alternative to comprehension.

Origins of N-reduction

Kreisel-Troelstra introduce the neighbourhood function principle NFP as a more constructive alternative to comprehension.

Definition (NFP)

For any formula $A(n^0)$, we have

$$(\forall f^1)(\exists n^0)A(\overline{f}n) \to (\exists \gamma \in K_0)(\forall f^1)A(\overline{f}\gamma(f)), \tag{1}$$

where '$\gamma \in K_0$' means that γ^1 is a (total) Kleene associate.

Origins of N-reduction

Kreisel-Troelstra introduce the neighbourhood function principle NFP as a more constructive alternative to comprehension.

Definition (NFP)

For any formula $A(n^0)$, we have

$$(\forall f^1)(\exists n^0)A(\overline{f}n) \to (\exists \gamma \in K_0)(\forall f^1)A(\overline{f}\gamma(f)), \tag{1}$$

where '$\gamma \in K_0$' means that γ^1 is a (total) Kleene associate.

Higher-order RM: most third-order theorems follow from (low-level) fragments of NFP. (TvD)

Origins of N-reduction

Kreisel-Troelstra introduce the neighbourhood function principle NFP as a more constructive alternative to comprehension.

Definition (NFP)

For any formula $A(n^0)$, we have

$$(\forall f^1)(\exists n^0)A(\overline{f}n) \to (\exists \gamma \in K_0)(\forall f^1)A(\overline{f}\gamma(f)), \tag{1}$$

where '$\gamma \in K_0$' means that γ^1 is a (total) Kleene associate.

Higher-order RM: most third-order theorems follow from (low-level) fragments of NFP. (TvD)

If NFP is restricted to A with type 2 parameters (only), then we can study it using N-reductions.

Existing frameworks

Existing frameworks

Takayuki Kihara and students: big inductive definition meant to capture theorems about all/many finite types (wip).

Existing frameworks

Takayuki Kihara and students: big inductive definition meant to capture theorems about all/many finite types (wip).

Makoto Fujiwara: explicit definition restricted to primitive recursive reductions.

Existing frameworks

Takayuki Kihara and students: big inductive definition meant to capture theorems about all/many finite types (wip).

Makoto Fujiwara: explicit definition restricted to primitive recursive reductions. Different goal: capture constructive reasoning via classical means.

Final Thoughts

Final Thoughts

The revolution is not an apple that falls when it is ripe. You have to make it fall. (AN & CG)

Final Thoughts

The revolution is not an apple that falls when it is ripe. You have to make it fall. (AN & CG)

We thank DFG, TU Darmstadt, John Templeton Foundation, and Alexander Von Humboldt Foundation for their generous support!

Final Thoughts

The revolution is not an apple that falls when it is ripe. You have to make it fall. (AN & CG)

We thank DFG, TU Darmstadt, John Templeton Foundation, and Alexander Von Humboldt Foundation for their generous support!

Thank you for your attention!

Final Thoughts

The revolution is not an apple that falls when it is ripe. You have to make it fall. (AN & CG)

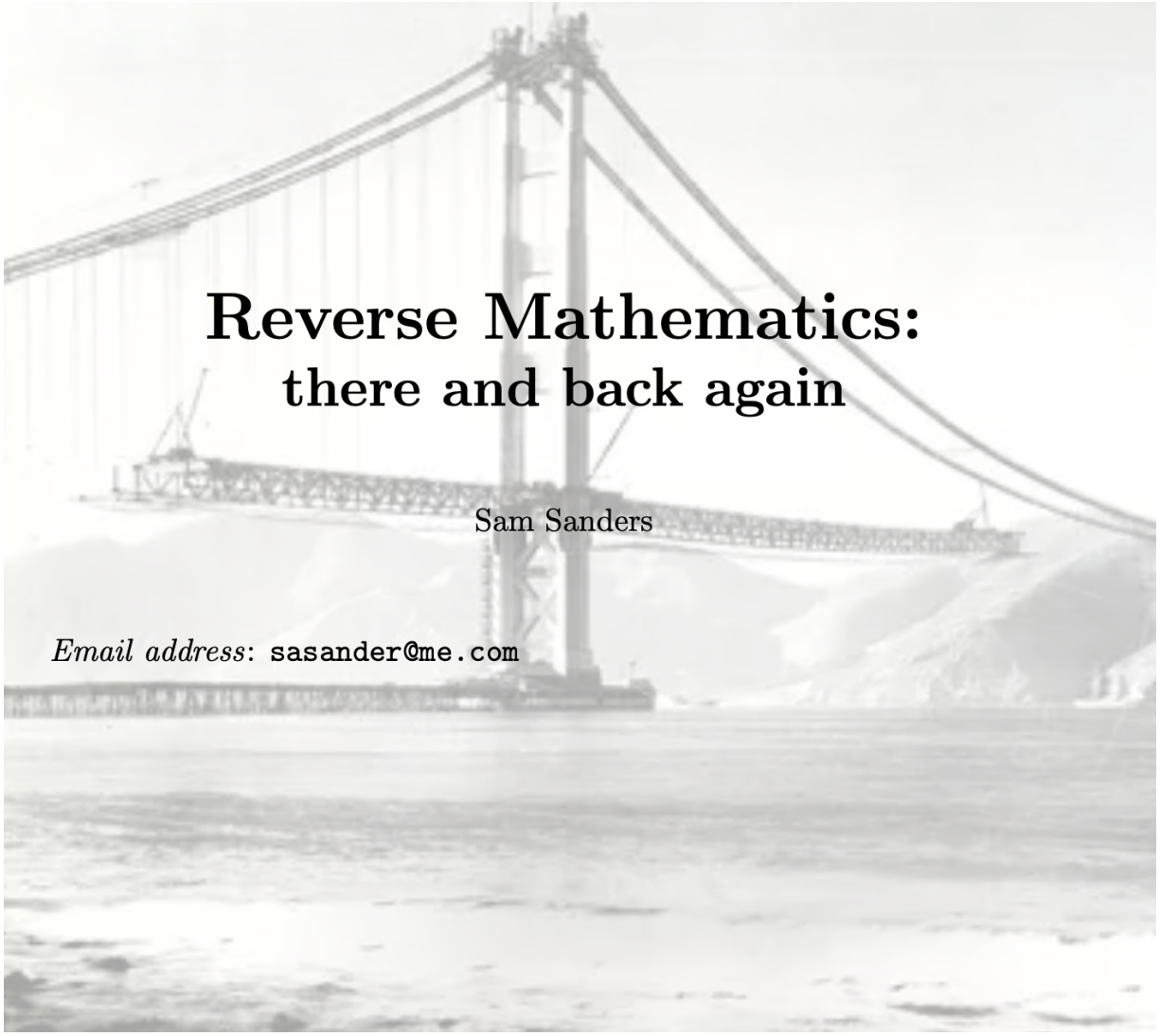
We thank DFG, TU Darmstadt, John Templeton Foundation, and Alexander Von Humboldt Foundation for their generous support!

Thank you for your attention!

Any questions?

And thanks to AN for all valuable advise, without which N-reduction would not have been discovered/invented.

Book on Reverse Mathematics



The details of N-reduction are in one of the final chapters.

Higher types redux-ish

Higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y, \Theta(Y)) \to (\forall Z^2)B(Z, \{e\}(\Theta, Z))\big] \qquad \text{(K)}$$

where $\{e\}(\Theta, Z)$ is a Kleene S1-S9 algorithm.

A version of (K) involving Turing machines:

$$(\forall Z^2, x^1)\big[A(t(Z), x) \to [\{e_0\}^{s(x,Z)}\downarrow \wedge B(Z, \{e_0\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e_0\}^y$ is a Turing machine with oracle y^1.

Higher types redux-ish

Higher-order computability theory studies whether:

$$(\forall \Theta^3)\big[(\forall Y^2)A(Y,\Theta(Y)) \to (\forall Z^2)B(Z,\{e\}(\Theta,Z))\big] \qquad \text{(K)}$$

where $\{e\}(\Theta, Z)$ is a Kleene S1-S9 algorithm.

A version of (K) involving Turing machines:

$$(\forall Z^2, x^1)\big[A(t(Z),x) \to [\{e_0\}^{s(x,Z)}\downarrow \wedge\, B(Z,\{e_0\}^{s(x,Z)})]\big], \qquad \text{(N)}$$

where t, s are primitive recursive and $\{e_0\}^y$ is a Turing machine with oracle y^1.

In case (N), we say that

solving the problem $(\forall Z^2)(\exists y^1)B(Z,x)$ *N*-reduces to solving the problem $(\forall Y^2)(\exists x^1)A(Y,x)$.