Synthetic Turing Reducibility in CIC

Yannick Forster[1,2] and Dominik Kirst[1]

Saarland University & Inria, Gallinette Team, Nantes

TYPES 2022, 21.06.2022

Yannick Forster received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101024493.

Synthetic Computability Theory

Work in constructive logic with the axiom that
partial functions on $\mathbb{N}$ are enumerable up to extensional equality, i.e.

$$\exists \theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N}). \forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\ \exists c. \theta_c \equiv f$$

Synthetic Computability Theory

Work in constructive logic with the axiom that
partial functions on $\mathbb{N}$ are enumerable up to extensional equality, i.e.

$$\exists \theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N}). \forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\ \exists c. \theta_c \equiv f$$

Consistent as a consequence of CT ("Church's thesis", c.f. [Kreisel, 1965] [Richman, 1983] [Forster, 2021])

Synthetic Computability Theory

Work in constructive logic with the axiom that
partial functions on $\mathbb{N}$ are enumerable up to extensional equality, i.e.

$$\exists \theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N}). \forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\ \exists c. \theta_c \equiv f$$

Consistent as a consequence of CT (“Church’s thesis”, c.f. [Kreisel, 1965] [Richman, 1983] [Forster, 2021])

Allows to do computability abstracted away from a model

Synthetic Computability Theory

Work in constructive logic with the axiom that
partial functions on $\mathbb{N}$ are enumerable up to extensional equality, i.e.

$$\exists \theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N}). \forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\ \exists c. \theta_c \equiv f$$

Consistent as a consequence of CT ("Church's thesis", c.f. [Kreisel, 1965] [Richman, 1983] [Forster, 2021])

Allows to do computability abstracted away from a model

Pioneered by Richman, Bridges, and Bauer:
The use of countable choice makes the law of excluded middle *false*

Synthetic Computability Theory in CIC

Work in CIC with the axiom that
partial functions on $\mathbb{N}$ are enumerable up to extensional equality, i.e.

$$\exists \theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N}). \forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\ \exists c. \theta_c \equiv f$$

Consistent as a consequence of CT ("Church's thesis", c.f. [Kreisel, 1965] [Richman, 1983] [Forster, 2021])

Allows to do computability abstracted away from a model

The law of excluded middle seems to be consistent
due to the absence of virually any (countable) choice principle

Synthetic Reducibility Theory

To define reducibility notions, just drop the word "computable":

$$A \preceq_m B := \exists f.\ \forall x.\ A(x) \leftrightarrow B(fx)$$

similar for one-one and truth-table reducibility

Synthetic Reducibility Theory

To define reducibility notions, just drop the word “computable”:

$$A \preceq_m B := \exists f.\ \forall x.\ A(x) \leftrightarrow B(fx)$$

similar for one-one and truth-table reducibility
jww Felix Jahn, Fabian Kunze, Nils Lauermann, Gert Smolka:

- Myhill Isomorphism theorem
- existence of Post’s simple predicates
- existence of Post’s hypersimple predicates [Forster et al., 2021]
- Non-random numbers (w.r.t. Kolmogorov complexity) are simple [Forster et al., 2022]

Synthetic Turing reductions

Predicate A on X is Turing-reducible to B on Y if

- there is $F : (Y \rightharpoonup \mathbb{B}) \to (X \rightharpoonup \mathbb{B})$
- such that F maps any given decider of B to a decider of A

Synthetic Turing reductions

Predicate A on X is Turing-reducible to B on Y if

- there is $F : (Y \rightharpoonup \mathbb{B}) \to (X \rightharpoonup \mathbb{B})$
- such that F maps any given decider of B to a decider of A

Theorem

If B is undecidable then for any A: $A \preceq_T B$

Synthetic Turing reductions

Predicate A on X is Turing-reducible to B on Y if

- there is $F : (Y \rightharpoonup \mathbb{B}) \to (X \rightharpoonup \mathbb{B})$
- $M : \{R : Y \to \mathbb{B} \to \mathbb{P} \mid R$ *functional*$\} \to \{R : X \to \mathbb{B} \to \mathbb{P} \mid R$ *functional*$\}$
- such that $\forall f.\ f$ *computes* $R \to F\,f$ *computes* $M\,R$
- such that M maps the characteristic relation of B to the one of A

Theorem

If B is undecidable then for any A: $A \preceq_T B$

Synthetic Turing reductions

Predicate A on X is Turing-reducible to B on Y if

- there is $F : (Y \rightharpoonup \mathbb{B}) \to (X \rightharpoonup \mathbb{B})$
- $M : \{R : Y \to \mathbb{B} \to \mathbb{P} \mid R \textit{ functional}\} \to \{R : X \to \mathbb{B} \to \mathbb{P} \mid R \textit{ functional}\}$
- such that $\forall f.\ f \textit{ computes } R \to F\,f \textit{ computes } M\,R$
- such that M maps the characteristic relation of B to the one of A

Theorem

If B is undecidable then for any A: $A \preceq_T B$

Proof.

Take $M\,R\,x\,b$ as: If R is computable, then false. If R is not computable, then b reflects $A\,x$. Factors through $F\,f\,x\,b :=$ false. □

Synthetic Turing reductions

Predicate A on X is Turing-reducible to B on Y if

- there is $F : (Y \rightharpoonup \mathbb{B}) \to (X \rightharpoonup \mathbb{B})$
- $M : \{R : Y \to \mathbb{B} \to \mathbb{P} \mid R$ *functional*$\} \to \{R : X \to \mathbb{B} \to \mathbb{P} \mid R$ *functional*$\}$
- such that $\forall f.\ f$ *computes* $R \to F\,f$ *computes* $M\,R$
- such that M maps the characteristic relation of B to the one of A
- and M is *continuous*

Synthetic Turing reductions

Predicate A on X is Turing-reducible to B on Y if

- there is $F : (Y \rightharpoonup \mathbb{B}) \to (X \rightharpoonup \mathbb{B})$
- $M : \{R : Y \to \mathbb{B} \to \mathbb{P} \mid R$ *functional*$\} \to \{R : X \to \mathbb{B} \to \mathbb{P} \mid R$ *functional*$\}$
- such that $\forall f.\ f$ *computes* $R \to F\,f$ *computes* $M\,R$
- such that M maps the characteristic relation of B to the one of A
- and M is *continuous*

$$M R x y \to \exists L : \mathbb{L} Y.\ L \subseteq \mathsf{Dom}(R) \land \forall R'.R \equiv_L R' \to M R' x y$$

idea due to Andrej Bauer [Bauer, 2020], observations about continuity of Turing reductions already by [Kleene, 1952], [Uspenskii, 1955], [Nerode, 1957], and [Davis, 1958].

Kleene-Post, Post's theorem, Post's problem

Theorem (Kleene-Post)

There are incomparable A and B below the halting problem.

Theorem (Post)

$A \in \Sigma^0_{n+1} \leftrightarrow \mathcal{S}_{\emptyset^{(n)}} A$ and $A \in \Sigma^0_{n+1} \to A \preceq_m \emptyset^{(n)}$.

Kleene-Post, Post's theorem, Post's problem

Theorem (Kleene-Post)

There are incomparable A and B below the halting problem.

Theorem (Post)

$A \in \Sigma^0_{n+1} \leftrightarrow \mathcal{S}_{\emptyset^{(n)}} A$ and $A \in \Sigma^0_{n+1} \rightarrow A \preceq_m \emptyset^{(n)}$.

At Thursday 12:15:

"Synthetic Versions of the Kleene-Post and Post's Theorem"

Dominik Kirst, Niklas Mück, Yannick Forster

Kleene-Post, Post's theorem, Post's problem

Theorem (Kleene-Post)

There are incomparable A and B below the halting problem.

Theorem (Post)

$A \in \Sigma^0_{n+1} \leftrightarrow \mathcal{S}_{\emptyset^{(n)}} A$ *and* $A \in \Sigma^0_{n+1} \to A \preceq_m \emptyset^{(n)}$.

At Thursday 12:15:
"Synthetic Versions of the Kleene-Post and Post's Theorem"
Dominik Kirst, Niklas Mück, Yannick Forster

Future work:

Theorem (Solution to Post's problem, Friedberg-Muchnik)

There is an enumerable but undecidable A with $A \not\preceq_T \emptyset'$.

Universal machine

In synthetic computability one works with a universal function

$$\theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N})$$

enumerating all partial functions up to equivalence.

To prove interesting theorems, one needs a continuous

$$\zeta : \mathbb{N} \to ((\mathbb{N} \rightharpoonup \mathbb{N}) \to (\mathbb{N} \rightharpoonup \mathbb{N}))$$

enumerating at least all continuous functionals.

Universal machine

In synthetic computability one works with a universal function

$$\theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N})$$

enumerating all partial functions up to equivalence.

To prove interesting theorems, one needs a continuous

$$\zeta : \mathbb{N} \to ((\mathbb{N} \rightharpoonup \mathbb{N}) \to (\mathbb{N} \rightharpoonup \mathbb{N}))$$

enumerating at least all continuous functionals.

Two lines of research:

- Try to construct ζ from θ with strongest possible specification
- Prove theorems first based on strong ζ and try to weaken specification as much as possible

Bauer, A. (2020).
Synthetic mathematics with an excursion into computability theory (slide set).
University of Wisconsin Logic seminar.

Davis, M. (1958).
Computability and Unsolvability.
McGraw-Hill.

Forster, Y. (2021).
Computability in Constructive Type Theory.
PhD thesis, Saarland University.

Forster, Y., Jahn, F., and Smolka, G. (2021).
A constructive and synthetic theory of reducibility: Myhill's isomorphism theorem and post's problem for many-one and truth-table reducibility in coq.
Pre-print.

Forster, Y., Kunze, F., and Lauermann, N. (2022).
Synthetic Kolmogorov Complexity in Coq.
working paper or preprint.

Kleene, S. C. (1952).
Introduction to metamathematics, volume 483.
van Nostrand New York.

Kreisel, G. (1965).
Mathematical logic.
Lectures in modern mathematics, 3:95–195.

Nerode, A. (1957).
General topology and partial recursive functionals.
Talks Cornell Summ. Inst. Symb. Log., pages 247–251.

Richman, F. (1983).
Church's thesis without tears.
The Journal of symbolic logic, 48(3):797–803.

Uspenskii, V. A. (1955).
On enumerable operators.
Dokl. Acad. Nauk., 103:773–776.