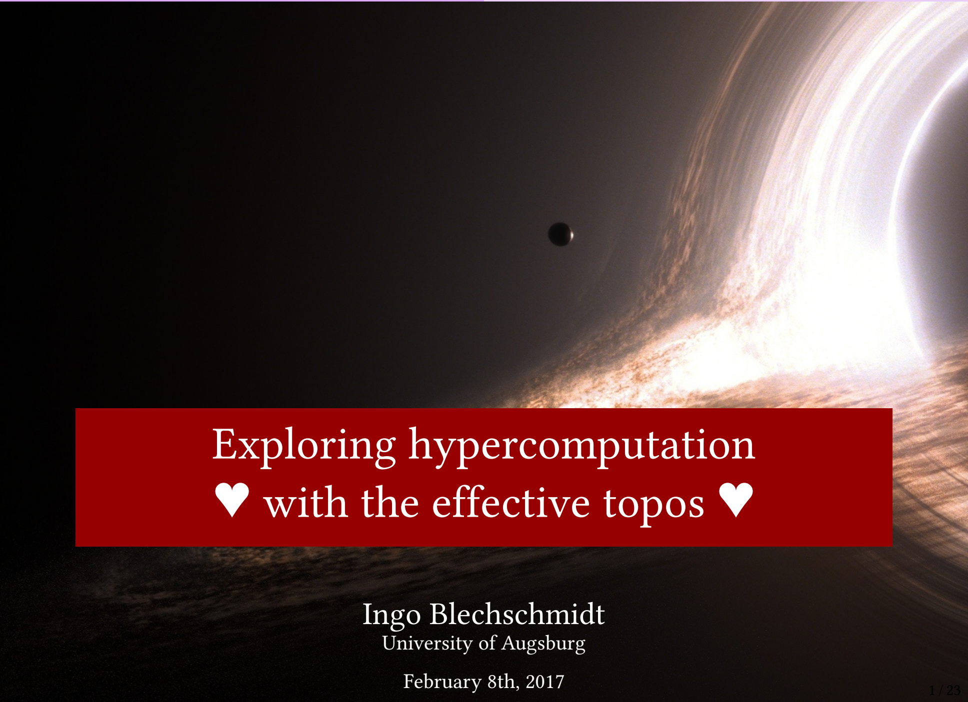


Exploring hypercomputation ♥ with the effective topos ♥

Ingo Blechschmidt
University of Augsburg

February 8th, 2017

1. Crash course on ordinal numbers

2. (Super) Turing machines
   - Bacis on Turing machines
   - Bacis on super Turing machines
   - The power of super Turing machines
   - Outlook on the bigger theory

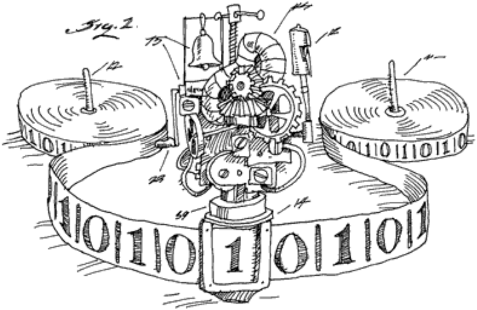
3. The effective topos
   - First steps in the effective topos
   - The wonder of constructive logic
   - Effective content of classical tautologies
   - Wrapping up

Part I

A crash course on ordinal numbers

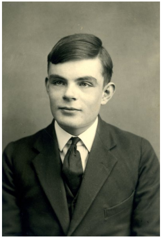
Part II

(Super) Turing machines

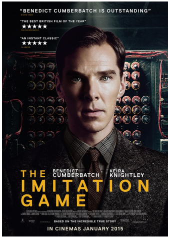


Basics on Turing machines

- Turing machines are idealised computers operating on an **infinite tape** according to a **finite list** of rules.
- The concept is astoundingly robust.
- A subset of $\mathbb{N}$ is **enumerable by a Turing machine** if and only if it's a Σ_1-set.



Alan Turing
(* 1912, † 1954)



worth seeing

Alison Bechdel
(* 1960)

Super Turing machines

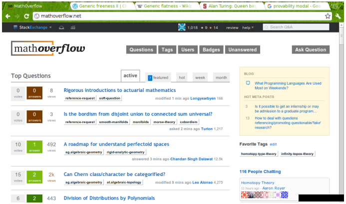
With super Turing machines, the time axis is more interesting:

- normal: $0, \; 1, \; 2, \; \ldots$
- super: $0, \; 1, \; 2, \; \ldots, \; \omega, \; \omega+1, \; \ldots, \; \omega \cdot 2, \; \omega \cdot 2 + 1, \; \ldots \ldots \ldots$

On reaching a limit ordinal time step like ω or $\omega \cdot 2$,

- the machine is put into a designated state,
- the read/write head is moved to the start of the tape, and
- the tape is set to the “lim sup” of all its previous contents.

Joel David Hamkins



MathOverflow

Andy Lewis

A question to you

What's the behaviour of this super Turing machine?

In the start state and the limit state, check whether the current cell contains a "1".

- If yes, then stop.
- If not, then flash that cell: set it to "1", then reset it to "0". Then unremittingly move the head rightwards.

A question to you

What's the behaviour of this super Turing machine?

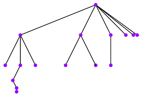
In the start state and the limit state, check whether the current cell contains a "1".

- If yes, then stop.
- If not, then flash that cell: set it to "1", then reset it to "0". Then unremittingly move the head rightwards.

Super Turing machines can break out of (some kinds of) infinite loops.

What can super Turing machines do?

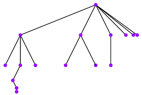
- Everything ordinary Turing machines can do.
- Verify number-theoretic statements.
- Decide whether a given ordinary Turing machine halts.
- Simulate super Turing machines.
- Decide Π^1_1- and Σ^1_1-statements:
  - "For any function $\mathbb{N} \to \mathbb{N}$ it holds that …"
  - "There is a function $\mathbb{N} \to \mathbb{N}$ such that …"



What can super Turing machines do?

- Everything ordinary Turing machines can do.
- Verify number-theoretic statements.
- Decide whether a given ordinary Turing machine halts.
- Simulate super Turing machines.
- Decide Π^1_1- and Σ^1_1-statements:
    - "For any function $\mathbb{N} \to \mathbb{N}$ it holds that …"
    - "There is a function $\mathbb{N} \to \mathbb{N}$ such that …"

But: Super Turing machines can't calculate all functions and can't write all 0/1-sequences to the tape.



Fun facts

- Any super Turing machine either halts or gets caught in an unbreakable infinite loop after **countably many steps**.
- An ordinal number α is **clockable** iff there is a super Turing machine which halts precisely after time step α.
  - Speed-up Lemma: If $\alpha + n$ is clockable, then so is α.
  - Big Gaps Theorem
  - Many Gaps Theorem
  - Gapless Blocks Theorem
- **Lost Melody Theorem:** There are 0/1-sequences which a super Turing machine can recognise, but not write to the tape.

Part III

The effective topos

The effective topos

- "$1 + 1 = 2$."
- "Any number is either prime or not."
- "Any function $\mathbb{N} \to \mathbb{N}$ is either the zero function or not."
- "Any function $\mathbb{N} \to \mathbb{N}$ is computable by a Turing machine."
- "Any function $\mathbb{R} \to \mathbb{R}$ is continuous."

The effective topos

- "$1 + 1 = 2$."
  True in Set, true in $\mathrm{Eff}(\mathrm{TM})$.
- "Any number is either prime or not."
- "Any function $\mathbb{N} \to \mathbb{N}$ is either the zero function or not."
- "Any function $\mathbb{N} \to \mathbb{N}$ is computable by a Turing machine."
- "Any function $\mathbb{R} \to \mathbb{R}$ is continuous."

The effective topos

- "$1 + 1 = 2$."
  True in Set, true in $\mathrm{Eff}(\mathrm{TM})$.

- "Any number is either prime or not."
  Trivially true in Set, nontrivially true in $\mathrm{Eff}(\mathrm{TM})$.

- "Any function $\mathbb{N} \to \mathbb{N}$ is either the zero function or not."

- "Any function $\mathbb{N} \to \mathbb{N}$ is computable by a Turing machine."

- "Any function $\mathbb{R} \to \mathbb{R}$ is continuous."

The effective topos

- "$1 + 1 = 2$."
  True in Set, true in Eff(TM).

- "Any number is either prime or not."
  Trivially true in Set, nontrivially true in Eff(TM).

- "Any function $\mathbb{N} \to \mathbb{N}$ is either the zero function or not."
  Trivially true in Set, false in Eff(TM).

- "Any function $\mathbb{N} \to \mathbb{N}$ is computable by a Turing machine."

- "Any function $\mathbb{R} \to \mathbb{R}$ is continuous."

The effective topos

- "$1 + 1 = 2$."
  True in Set, true in $\mathrm{Eff}(\mathrm{TM})$.
- "Any number is either prime or not."
  Trivially true in Set, nontrivially true in $\mathrm{Eff}(\mathrm{TM})$.
- "Any function $\mathbb{N} \to \mathbb{N}$ is either the zero function or not."
  Trivially true in Set, false in $\mathrm{Eff}(\mathrm{TM})$.
- "Any function $\mathbb{N} \to \mathbb{N}$ is computable by a Turing machine."
  False in Set, trivially true in $\mathrm{Eff}(\mathrm{TM})$.
- "Any function $\mathbb{R} \to \mathbb{R}$ is continuous."

The effective topos

- "$1 + 1 = 2$."
  True in Set, true in $\mathrm{Eff}(\mathrm{TM})$.

- "Any number is either prime or not."
  Trivially true in Set, nontrivially true in $\mathrm{Eff}(\mathrm{TM})$.

- "Any function $\mathbb{N} \to \mathbb{N}$ is either the zero function or not."
  Trivially true in Set, false in $\mathrm{Eff}(\mathrm{TM})$.

- "Any function $\mathbb{N} \to \mathbb{N}$ is computable by a Turing machine."
  False in Set, trivially true in $\mathrm{Eff}(\mathrm{TM})$.

- "Any function $\mathbb{R} \to \mathbb{R}$ is continuous."
  False in Set, true in $\mathrm{Eff}(\mathrm{TM})$.

First steps in the effective topos

$\mathrm{Eff}(\mathrm{TM}) \models$ "For any number n there is a prime $p > n$."

means:

There is a Turing machine which reads a number n as input and outputs a prime number $p > n$.

"Realisability Theory"

First steps in the effective topos

$\mathrm{Eff}(\mathrm{TM}) \models$ "For any number n there is a prime $p > n$."

means:

There is a Turing machine which reads a number n as input and outputs a prime number $p > n$.

$\mathrm{Eff}(\mathrm{TM}) \models$ "Any number has a prime factor decomposition."

means:

There is a Turing machine which reads a number n as input and outputs a list of primes, the product of which is n.

First steps in the effective topos

$\mathrm{Eff}(\mathrm{TM}) \models$ "For any number n there is a prime $p > n$."

means:

There is a Turing machine which reads a number n as input and outputs a prime number $p > n$.

$\mathrm{Eff}(\mathrm{TM}) \models$ "Any number has a prime factor decomposition."

means:

There is a Turing machine which reads a number n as input and outputs a list of primes, the product of which is n.

$\mathrm{Eff}(\mathrm{TM}) \models$ "Any number is either prime or not prime."

means:

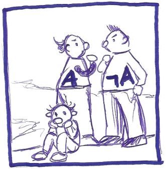
There is a Turing machine which reads a number n as input and outputs YES or NO depending on whether n is prime or not.

What's true in alternate toposes?

Metatheorem: If a statement has a **constructive proof**, then it holds in **any topos**.

Constructive logic is like classical logic, except we don't suppose the **law of excluded middle** (LEM), which says:

- "Any statement is either true or not true."
- "If a statement is *not not* true, then it's true."



Nonconstructive proofs

Theorem. There are **irrational** numbers x und y such that x^y is rational.

Nonconstructive proofs

Theorem. There are **irrational** numbers x und y such that x^y is rational.

Proof. Either $\sqrt{2}^{\sqrt{2}}$ is rational or not.

1. In the first case we are done.
2. In the second case we set $x := \sqrt{2}^{\sqrt{2}}$ and $y := \sqrt{2}$. Then $x^y = \sqrt{2}^{\sqrt{2}\cdot\sqrt{2}} = \sqrt{2}^2 = 2$ is rational.

Appreciating constructive logic

At first sight, dropping the axiom of excluded middle looks like a sad thing to do. It's a useful axiom! However:

- The axiom is not needed as often as one would think.
- The abstinence is good for your mental hygiene.
- Constructive logic allows for finer distinctions.
- From constructive proofs one can mechanically extract programs which witness the proved statements.
- **Dropping the law of excluded middle allows to add curious unconvential axioms.**

LEM for equality of functions

$\mathrm{Eff}(\mathrm{TM}) \models$ "Any function $f : \mathbb{N} \to \mathbb{N}$ is either the zero function or not."

means:

There is a Turing machine which reads the source of a Turing machine M, which calculates a function $\mathbb{N} \to \mathbb{N}$, as input, and finds out whether M always yields zero or not.

That's false.

LEM for equality of functions

$\mathrm{Eff}(\mathrm{TM}) \models$ "Any function $f : \mathbb{N} \to \mathbb{N}$ is either the zero function or not."

means:

There is a Turing machine which reads the source of a Turing machine M, which calculates a function $\mathbb{N} \to \mathbb{N}$, as input, and finds out whether M always yields zero or not.

That's false.

The statement is true in $\mathrm{Eff}(\mathrm{STM})$, the effective topos associated to super Turing machines.

LEM for the halting problem

$\mathrm{Eff}(\mathrm{TM}) \models$ "Any Turing machine halts or doesn't halt."

means:

There is a Turing machine which reads the source of a Turing machine M as input and finds out whether M halts or not.

That's false.

LEM for the halting problem

$\mathrm{Eff}(\mathrm{TM}) \models$ "Any Turing machine halts or doesn't halt."

means:

There is a Turing machine which reads the source of a Turing machine M as input and finds out whether M halts or not.

That's false.

The statement is true in $\mathrm{Eff}(\mathrm{STM})$.

LEM for equality of real numbers

The statement

"Every real number is either zero or not zero."

is in constructive logic equivalent to

"Every Turing machine halts or doesn't halt.",

so it's not true in Eff(TM), but in Eff(STM).

LEM for equality of real numbers

The statement

"Every real number is either zero or not zero."

is in constructive logic equivalent to

"Every Turing machine halts or doesn't halt.",

so it's not true in $\mathrm{Eff}(\mathrm{TM})$, but in $\mathrm{Eff}(\mathrm{STM})$.

For a Turing machine M consider the real number $0.000\ldots$ whose n'th decimal digit is a one iff M halts after step n.

For a real number x consider the Turing machine which searches the digits of x for a nonzero digit.

Markov's principle

$\mathrm{Eff}(\mathrm{TM}) \models$ "For any function $f : \mathbb{N} \to \mathbb{N}$ which is not the zero function, there is a number $n \in \mathbb{N}$ such that $f(n) \neq 0$."

means:

> There is a Turing machine which reads the source of a Turing machine M, which calculates a function $\mathbb{N} \to \mathbb{N}$ which is not the zero function, as input and outputs a number n such that $M(n)$ is not zero.

Markov's principle

$\mathrm{Eff}(\mathrm{TM}) \models$ "For any function $f : \mathbb{N} \to \mathbb{N}$ which is not the zero function, there is a number $n \in \mathbb{N}$ such that $f(n) \neq 0$."

means:

There is a Turing machine which reads the source of a Turing machine M, which calculates a function $\mathbb{N} \to \mathbb{N}$ which is not the zero function, as input and outputs a number n such that $M(n)$ is not zero.

That's true! By unbounded search.

Searching uncountable sets

> "For any function $f : \mathbb{N} \to \mathbb{N}$ from numbers to numbers, there either exists a number n such that $f(n)$ is one or there is no such number."

This statement is false in $\mathrm{Eff}(\mathrm{TM})$.

> "For any function $P : L(\mathbb{B}) \to \mathbb{B}$ from infinite lists of booleans to booleans, there either exists a list x such that $P(x)$ is true or there is no such list."

Searching uncountable sets

> "For any function $f : \mathbb{N} \to \mathbb{N}$ from numbers to numbers, there either exists a number n such that $f(n)$ is one or there is no such number."

This statement is false in $\mathrm{Eff}(\mathrm{TM})$.

> "For any function $P : L(\mathbb{B}) \to \mathbb{B}$ from infinite lists of booleans to booleans, there either exists a list x such that $P(x)$ is true or there is no such list."

This statement is true in $\mathrm{Eff}(\mathrm{TM})$!

The Church–Turing thesis

The **Church–Turing thesis** states:

> If a function $f : \mathbb{N} \to \mathbb{N}$ is calculable in the real world, then it's also calculable by a Turing machine.

$\mathrm{Eff}(\mathrm{TM}) \models$ "Any function $f : \mathbb{N} \to \mathbb{N}$ is calculable by a Turing machine."

means:

> There is a Turing machine which reads the source of a Turing machine calculating a function $f : \mathbb{N} \to \mathbb{N}$ as input and outputs the source of a Turing machine which calculates f.
>
> That's trivial, echo the input back to the user.

The Church–Turing thesis

The **Church–Turing thesis** states:

> If a function $f : \mathbb{N} \to \mathbb{N}$ is calculable in the real world, then it's also calculable by a Turing machine.

$\mathrm{Eff}(\mathrm{TM}) \models$ "Any function $f : \mathbb{N} \to \mathbb{N}$ is calculable by a Turing machine."

means:

> There is a Turing machine which reads the source of a Turing machine calculating a function $f : \mathbb{N} \to \mathbb{N}$ as input and outputs the source of a Turing machine which calculates f.
>
> That's trivial, echo the input back to the user.

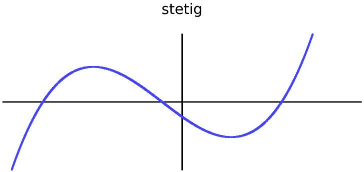
In $\mathrm{Eff}(\mathrm{STM})$ the statement is false.

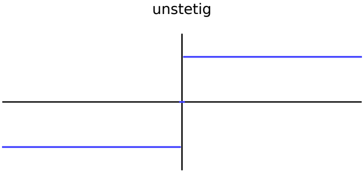
Automatic continuity

The following statement is wildly **false** in Set:

"Every function $f : \mathbb{R} \to \mathbb{R}$ is continuous."

A function f is **continuous** if and only if, for calculating $f(x)$ to finitely many digits, finitely many digits of x suffice.



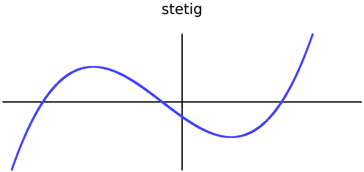


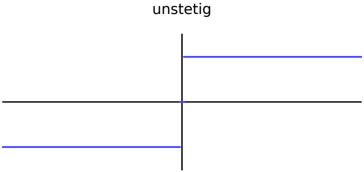
Automatic continuity

The following statement is wildly **false** in Set:

"Every function $f : \mathbb{R} \to \mathbb{R}$ is continuous."

A function f is **continuous** if and only if, for calculating $f(x)$ to finitely many digits, finitely many digits of x suffice.





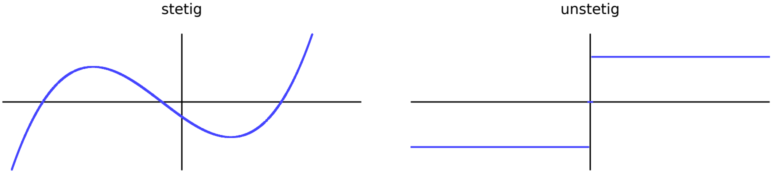
True in $\mathrm{Eff}(\mathrm{TM})$.

Automatic continuity

The following statement is wildly **false** in Set:

"Every function $f : \mathbb{R} \to \mathbb{R}$ is continuous."

A function f is **continuous** if and only if, for calculating $f(x)$ to finitely many digits, finitely many digits of x suffice.



True in $\mathrm{Eff}(\mathrm{TM})$. True in $\mathrm{Eff}(\mathrm{RW})$, if private communication channels are possible and only finitely many computational steps can be executed in finite time.

Wrapping up

- Effective toposes are a good vehicle for studying the nature of computation.
- The effective toposes build links between constructive mathematics and programming.
- Toposes allow for curious dream axioms.
- Toposes also have a geometric flavour: points, subtoposes, maps between toposes.

Wrapping up

- Effective toposes are a good vehicle for studying the nature of computation.
- The effective toposes build links between constructive mathematics and programming.
- Toposes allow for curious dream axioms.
- Toposes also have a geometric flavour: points, subtoposes, maps between toposes.

There is more to mathematics than the standard topos.