Computing Over Generalisations of the Reals

Lorenzo Galeotti
5th Workshop on Generalised Baire Spaces
Bristol, 4 February 2020

Amsterdam University College
Excellence and Diversity in a Global City

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

The Real Line

The real line has a central role both in mathematics and in computability theory.

Because of the complex topological and combinatorial structure of $\mathbb{R}$ often other better behaved spaces such as the Cantor space 2^ω are used in order to study properties of $\mathbb{R}$.

$$\mathbb{R} \xleftrightarrow{\text{Transfer}} 2^\omega$$

Transfinite Computability

Surreal Numbers

Generalised Real Line

Generalised Type 2 Computability

Surreal Blum-Shub-Smale machines

Computations Over the Reals via Tapes

Cantor space is used to induce notions of computability over spaces of size 2^ω.

Transfinite Computability

Surreal Numbers

Generalised Real Line

Generalised Type 2 Computability

Surreal Blum-Shub-Smale machines

Computations Over the Reals via Tapes

Cantor space is used to induce notions of computability over spaces of size 2^ω.

Type 2 Turing Machines (T2TM) are a model of computation whose hardware is essentially the same of that of classical Turing machines. Contrary to classical Turing Machines T2TM are allowed to run for infinite (ω) many steps. The result of the computation is then taken to be the limit of the content of the output tape of the T2TM.

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Computations Over the Reals via Tapes

Cantor space is used to induce notions of computability over spaces of size 2^ω.

Type 2 Turing Machines (T2TM) are a model of computation whose hardware is essentially the same of that of classical Turing machines. Contrary to classical Turing Machines T2TM are allowed to run for infinite (ω) many steps. The result of the computation is then taken to be the limit of the content of the output tape of the T2TM.

These machines induce a notion of computability over Cantor space. Using coding functions one can then transfer this notion to other spaces, e.g., the real line.

Computations Over the Reals via Registers

A different approach to computability over the reals is that of register machines.

Computations Over the Reals via Registers

A different approach to computability over the reals is that of register machines.

Blum-Shub-Smale machines (BSS machines) work on registers. Each register contains a real number. The machine is only allowed to run for a finite amount of time. At each step the machine can:

- test the content of a register and perform a jump based on the result;
- apply a rational function to registers.

Computations Over the Reals via Registers

A different approach to computability over the reals is that of register machines.

Blum-Shub-Smale machines (BSS machines) work on registers. Each register contains a real number. The machine is only allowed to run for a finite amount of time. At each step the machine can:

- test the content of a register and perform a jump based on the result;
- apply a rational function to registers.

In this approach no coding is needed and the notion of computability is unique.

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Computability, Space and Time

In classical computability theory computations are thought as finite and *discrete* processes carried out by (idealised) machines with unbounded finite data.

Computability, Space and Time

In classical computability theory computations are thought as finite and *discrete* processes carried out by (idealised) machines with unbounded finite data.

In defining notions of computability over the reals the assumptions on space and time are relaxed.

Model	Basic Data	Space	Time
T2TM	2^ω	ω	ω
BSS	$\mathbb{R}$	ω	finite

Classical Transfinite Computability

The idea of *transfinite computability* is to allow computations to “go on forever”, i.e., for a transfinite amount of time.

Classical Transfinite Computability

The idea of *transfinite computability* is to allow computations to "go on forever", i.e., for a transfinite amount of time.

- **Infinite Time Turing Machines (ITTM)**: Introduced by Hamkins and Lewis, have the same hardware of normal Turing machine but are allowed to carry out a transfinite number of steps.
- **Ordinal Turing Machines (OTM)**: Introduced by Koepke, these machines have tapes of transfinite length and are allowed to run for a transfinite number of steps.

Model	Space	Time
ITTM	ω	transfinite
OTM	transfinite	transfinite

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Generalising Computability: The Problem

How can the classical computability over the reals be generalised?

Generalising Computability: The Problem

How can the classical computability over the reals be generalised?

What is the right generalisation of $\mathbb{R}$ in this context?

Generalising Computability: The Problem

How can the classical computability over the reals be generalised?

What is the right generalisation of $\mathbb{R}$ in this context?

In the past few years there were essentially two generalisations of $\mathbb{R}$ in the context of generalised descriptive set theory.

- The long reals κ-$\mathbb{R}$: invented by Sikorski in the 60s and recently studied by Asperó and Tsaprounis.
- The generalised reals $\mathbb{R}_\kappa$: used in my Master's thesis in the context of generalised DST and generalised computable analysis.

Surreal numbers

Surreal numbers were introduced by Conway in order to formalise the abstract notion of *number*:

Definition (Surreal numbers)

A surreal number is a function from an ordinal $\alpha \in \mathrm{On}$ to $\{+,-\}$ (i.e., a sequence of pluses and minuses of ordinal length).

- We will denote the class of surreal numbers by No.
- Given an ordinal α we denote the set of surreals of length $<\alpha$ by $\mathrm{No}_{<\alpha}$.
- Surreal numbers are naturally ordered using the lexicographic order.

Simplicity Theorem

One of the most important results in the basic theory of surreal numbers is the Simplicity Theorem:

Theorem (Simplicity Theorem)

Let L and R be two sets of surreal numbers such that $L < R$. Then there is a unique surreal z, denoted by $[L \mid R]$, of minimal length such that $L < \{z\} < R$. We will call $[L \mid R]$ a representation of z.

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Simplicity Theorem

One of the most important results in the basic theory of surreal numbers is the Simplicity Theorem:

Theorem (Simplicity Theorem)

Let L and R be two sets of surreal numbers such that $L < R$. Then there is a unique surreal z, denoted by $[L \mid R]$, of minimal length such that $L < \{z\} < R$. We will call $[L \mid R]$ a representation of z.

Canonical Representation of x:

- L: set y such that $y+$ is a prefix of x.
- R: set y such that $y-$ is a prefix of x.

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Simplicity Theorem

One of the most important results in the basic theory of surreal numbers is the Simplicity Theorem:

Theorem (Simplicity Theorem)

Let L and R be two sets of surreal numbers such that $L < R$. Then there is a unique surreal z, denoted by $[L \mid R]$, of minimal length such that $L < \{z\} < R$. We will call $[L \mid R]$ a representation of z.

Canonical Representation of x:

- L: set y such that $y+$ is a prefix of x.
- R: set y such that $y-$ is a prefix of x.

For example:

- $\langle\rangle \mapsto [\emptyset \mid \emptyset]$.

Simplicity Theorem

One of the most important results in the basic theory of surreal numbers is the Simplicity Theorem:

Theorem (Simplicity Theorem)

Let L and R be two sets of surreal numbers such that $L < R$. Then there is a unique surreal z, denoted by $[L \mid R]$, of minimal length such that $L < \{z\} < R$. We will call $[L \mid R]$ a representation of z.

Canonical Representation of x:

- L: set y such that $y+$ is a prefix of x.
- R: set y such that $y-$ is a prefix of x.

For example:

- $\langle\rangle \mapsto [\emptyset \mid \emptyset]$.
- $+ \mapsto [\langle\rangle \mid \emptyset]$.

Simplicity Theorem

One of the most important results in the basic theory of surreal numbers is the Simplicity Theorem:

Theorem (Simplicity Theorem)

Let L and R be two sets of surreal numbers such that $L < R$. Then there is a unique surreal z, denoted by $[L \mid R]$, of minimal length such that $L < \{z\} < R$. We will call $[L \mid R]$ a representation of z.

Canonical Representation of x:

- L: set y such that $y+$ is a prefix of x.
- R: set y such that $y-$ is a prefix of x.

For example:

- $\langle\rangle \mapsto [\emptyset \mid \emptyset]$.
- $+ \mapsto [\langle\rangle \mid \emptyset]$.
- $\underbrace{+\dots}_{\omega \text{ times}} - \mapsto [\langle\rangle, +, ++, +++, \dots \mid \underbrace{+\dots}_{\omega \text{ times}}]$.

Surreal Numbers Operations

Definition (Surreal Sum)

Let x and y be two surreal numbers and $[L_x \mid R_x]$, $[L_y \mid R_y]$ be their canonical representations. Then we define the sum $x +_s y$ as follows:

$$x +_s y = [x_L +_s y, x +_s y_L \mid x_R +_s y, x +_s y_R]$$

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Surreal Numbers Operations

Definition (Surreal Sum)

Let x and y be two surreal numbers and $[L_x \mid R_x]$, $[L_y \mid R_y]$ be their canonical representations. Then we define the sum $x +_s y$ as follows:

$$x +_s y = [x_L +_s y, x +_s y_L \mid x_R +_s y, x +_s y_R]$$

where $x_L \in L_x$, $_L \in L_y$, $x_R \in R_x$ and $y_R \in R_y$.

Surreal Numbers Operations

Definition (Surreal Sum)

Let x and y be two surreal numbers and $[L_x \mid R_x]$, $[L_y \mid R_y]$ be their canonical representations. Then we define the sum $x +_s y$ as follows:

$$x +_s y = [x_L +_s y, x +_s y_L \mid x_R +_s y, x +_s y_R]$$

where $x_L \in L_x$, $_L \in L_y$, $x_R \in R_x$ and $y_R \in R_y$.

This is just demanding that:

- $\forall x_L \in L_x x_L +_s y < x +_s y$,
- $\forall x_R \in R_x x_R +_s y > x +_s y$,
- $\forall y_L \in L_y x +_s y_L < x +_s y$,
- $\forall y_R \in R_y x +_s y_R > x +_s y$.

Namely $+_s$ should make No into an ordered additive group.

The surreal numbers as a RCF

In his introduction of surreal numbers Conway proved:

Theorem

The surreal numbers form a real closed field *.

Later Ehrlich proved that:

Theorem

No *is the unique homogeneous universal real closed field.*

Theorem

Every real closed field is isomorphic to an initial subfield of No.

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Generalising $\mathbb{R}$

We start from a cardinal $\kappa > \omega$ such that:

$$\kappa^{<\kappa} = \kappa$$

We want to define $\mathbb{R}_\kappa$ such that:

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Generalising $\mathbb{R}$

We start from a cardinal $\kappa > \omega$ such that:

$$\kappa^{<\kappa} = \kappa$$

We want to define $\mathbb{R}_\kappa$ such that:

- $|\mathbb{R}_\kappa| = 2^\kappa$.

Generalising $\mathbb{R}$

We start from a cardinal $\kappa > \omega$ such that:

$$\kappa^{<\kappa} = \kappa$$

We want to define $\mathbb{R}_\kappa$ such that:

- $|\mathbb{R}_\kappa| = 2^\kappa$.
- $\mathbb{R}_\kappa$ has to be a real closed field extension of $\mathbb{R}$.

Generalising $\mathbb{R}$

We start from a cardinal $\kappa > \omega$ such that:

$$\kappa^{<\kappa} = \kappa$$

We want to define $\mathbb{R}_\kappa$ such that:

- $|\mathbb{R}_\kappa| = 2^\kappa$.
- $\mathbb{R}_\kappa$ has to be a real closed field extension of $\mathbb{R}$.
- $\mathbb{R}_\kappa$ has a dense subset of cardinality κ (i.e., $\mathbb{R}_\kappa$ has weight κ).

Generalising $\mathbb{R}$

We start from a cardinal $\kappa > \omega$ such that:

$$\kappa^{<\kappa} = \kappa$$

We want to define $\mathbb{R}_\kappa$ such that:

- $|\mathbb{R}_\kappa| = 2^\kappa$.
- $\mathbb{R}_\kappa$ has to be a real closed field extension of $\mathbb{R}$.
- $\mathbb{R}_\kappa$ has a dense subset of cardinality κ (i.e., $\mathbb{R}_\kappa$ has weight κ).
- $\mathbb{R}_\kappa$ is Cauchy complete.

Generalising $\mathbb{R}$

We start from a cardinal $\kappa > \omega$ such that:

$$\kappa^{<\kappa} = \kappa$$

We want to define $\mathbb{R}_\kappa$ such that:

- $|\mathbb{R}_\kappa| = 2^\kappa$.
- $\mathbb{R}_\kappa$ has to be a real closed field extension of $\mathbb{R}$.
- $\mathbb{R}_\kappa$ has a dense subset of cardinality κ (i.e., $\mathbb{R}_\kappa$ has weight κ).
- $\mathbb{R}_\kappa$ is Cauchy complete.
- We want to be able to prove some classical theorems from real analysis for $\mathbb{R}_\kappa$, e.g., IVT, BWT, HBT etc.

The generalised real line $\mathbb{R}_\kappa$

The real closed field $\mathrm{No}_{<\kappa}$ has almost all the properties we want.

The generalised real line $\mathbb{R}_\kappa$

The real closed field $\mathrm{No}_{<\kappa}$ has almost all the properties we want.

- $|\mathrm{No}_{<\kappa}| = \kappa$,

The generalised real line $\mathbb{R}_\kappa$

The real closed field $\mathrm{No}_{<\kappa}$ has almost all the properties we want.

- $|\,\mathrm{No}_{<\kappa}\,| = \kappa$,
- $\mathrm{No}_{<\kappa}$ is not Cauchy complete.

The generalised real line $\mathbb{R}_\kappa$

The real closed field $\mathrm{No}_{<\kappa}$ has almost all the properties we want.

- $|\mathrm{No}_{<\kappa}| = \kappa$,
- $\mathrm{No}_{<\kappa}$ is not Cauchy complete.

We define $\mathbb{R}_\kappa$ as the Cauchy completion of $\mathrm{No}_{<\kappa}$. We will denote $\mathrm{No}_{<\kappa}$ by $\mathbb{Q}_\kappa$, and call them κ-rational numbers.

The generalised real line $\mathbb{R}_\kappa$

The real closed field $\mathrm{No}_{<\kappa}$ has almost all the properties we want.

- $|\mathrm{No}_{<\kappa}| = \kappa$,
- $\mathrm{No}_{<\kappa}$ is not Cauchy complete.

We define $\mathbb{R}_\kappa$ as the Cauchy completion of $\mathrm{No}_{<\kappa}$. We will denote $\mathrm{No}_{<\kappa}$ by $\mathbb{Q}_\kappa$, and call them κ-rational numbers.

Theorem (G., CiE 2016)

The set $\mathbb{R}_\kappa$ is the unique Cauchy-complete real closed field with the following properties:

1. $|\mathbb{R}_\kappa| = 2^\kappa$.
2. $\mathbb{R}_\kappa$ *is* κ*-saturated.*
3. $\mathrm{Cof}(\mathbb{R}_\kappa) = \mathrm{Coi}(\mathbb{R}_\kappa) = w(\mathbb{R}_\kappa) = \kappa$.

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Analysis over $\mathbb{R}_\kappa$

The field $\mathbb{R}_\kappa$ behaves quite well with respect to real analysis.

Analysis over $\mathbb{R}_\kappa$

The field $\mathbb{R}_\kappa$ behaves quite well with respect to real analysis.

1) [G. CiE 2016] Since $\mathbb{R}_\kappa$ is κ-saturated a version of IVT and EVT.

Analysis over $\mathbb{R}_\kappa$

The field $\mathbb{R}_\kappa$ behaves quite well with respect to real analysis.

1) [G. CiE 2016] Since $\mathbb{R}_\kappa$ is κ-saturated a version of IVT and EVT.

2) [Carl,G.,Löwe 2019] κ has the tree property iff a natural ([G., Hanafi, Löwe]) weakening of BWT can be proved to hold on $\mathbb{R}_\kappa$ despite it is κ-saturated.

Generalising Type 2 Computability

In a joint paper with Hugo Nobrega (CiE 2017) OTMs to induce a notion of computability over surreal numbers.

The idea was to follow the classical construction of T2TMs.

Generalising Type 2 Computability

In a joint paper with Hugo Nobrega (CiE 2017) OTMs to induce a notion of computability over surreal numbers.

The idea was to follow the classical construction of T2TMs.

In particular one can consider OTMs with bounded tape length and time which are allowed to run for κ-many steps and whose output tape is write-only.

Generalising Type 2 Computability

In a joint paper with Hugo Nobrega (CiE 2017) OTMs to induce a notion of computability over surreal numbers.

The idea was to follow the classical construction of T2TMs.

In particular one can consider OTMs with bounded tape length and time which are allowed to run for κ-many steps and whose output tape is write-only.

This is a natural modifications of the κ-Turing Machines considered by Koepke & Seyfferth.

Generalising Type 2 Computability on $\mathbb{R}$

As in the classical case, we introduced codings of $\mathbb{R}_\kappa$ into 2^κ. In particular we used the following three codings:

- Fast Cauchy Sequences Representation $\delta_{\mathbb{R}_\kappa}$: every real is represented by a κ-sequence of elements of $\mathbb{Q}_\kappa$ converging to it at a fixed rate.
- Veronese cut Representation $\delta^{\mathrm{V}}_{\mathbb{R}_\kappa}$: every real is represented by a cut $[L|R]$ with L and R getting infinitely close to each other.

With these codings surreal operations are computable by these machines.

Theorem.[G., Nobrega] The fields operations of the surreal numbers are $\delta^{\mathrm{v}}_{\mathbb{R}_\kappa}$-computable.

Generalising Type 2 Computability on $\mathbb{R}$

As in the classical case, we introduced codings of $\mathbb{R}_\kappa$ into 2^κ. In particular we used the following three codings:

- Fast Cauchy Sequences Representation $\delta_{\mathbb{R}_\kappa}$: every real is represented by a κ-sequence of elements of $\mathbb{Q}_\kappa$ converging to it at a fixed rate.
- Veronese cut Representation $\delta^{\mathrm{V}}_{\mathbb{R}_\kappa}$: every real is represented by a cut $[L|R]$ with L and R getting infinitely close to each other.

With these codings surreal operations are computable by these machines.

Theorem.[G., Nobrega] The fields operations of the surreal numbers are $\delta^{\mathrm{v}}_{\mathbb{R}_\kappa}$-computable.

This result is the first step towards a generalisation of the classical framework of computable analysis to uncountable cardinals.

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Generalising Blum-Shub-Smale machines

A generalisation of BSS machines was proposed by Koepke and Seyfferth (CiE 2012).
BSS are asymmetric in the sense that space is infinite but time is finite.

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Generalising Blum-Shub-Smale machines

A generalisation of BSS machines was proposed by Koepke and Seyfferth (CiE 2012).
BSS are asymmetric in the sense that space is infinite but time is finite. In their work Koepke and Seyfferth introduced a transfinite version of BSS machines called *Infinite Time BSS* machines (ITBM) which still work on real numbers, but which are allowed to run for a transfinite amount of steps.
Since they are allowed to run for transfinite time on registers containing real numbers, ITBM are a generalisation of BSS machines analogous to ITTMs.

Model	Space	Time
ITTM	ω	transfinite
OTM	transfinite	transfinite
ITBM	ω	transfinite

ITBM Computational Power

While in some sense ITBMs are a generalisation of BSS machines that is analogous to that of ITTMs it is worth to note that ITBMs are a quite weak model of computation.

ITBM Computational Power

While in some sense ITBMs are a generalisation of BSS machines that is analogous to that of ITTMs it is worth to note that ITBMs are a quite weak model of computation.

Koepke & Morozov showed that the ITBM computable reals are exactly those in L_{ω^ω}.

ITBM Computational Power

While in some sense ITBMs are a generalisation of BSS machines that is analogous to that of ITTMs it is worth to note that ITBMs are a quite weak model of computation.

Koepke & Morozov showed that the ITBM computable reals are exactly those in L_{ω^ω}.

This is quite below the usual model of transfinite computation, e.g., ITRMs, ITTMs, OTMs.

ITBM Computational Power

While in some sense ITBMs are a generalisation of BSS machines that is analogous to that of ITTMs it is worth to note that ITBMs are a quite weak model of computation.

Koepke & Morozov showed that the ITBM computable reals are exactly those in L_{ω^ω}.

This is quite below the usual model of transfinite computation, e.g., ITRMs, ITTMs, OTMs.

In an ongoing project with Carl we are studying a strengthening of ITBMs in which the machine uses $\liminf$ instead of just Cauchy limits at limit stages. So far we have proved that these machines are much stronger than ITRM and we provided an (unfortunately big) upper bound of for their strength (the first Σ_2-admissible).

Generalising Blum-Shub-Smale machines

A symmetric version of transfinite Blum-Shub-Smale machine is still missing.

	Tapes	Registers	Space	Time
Asym.	ITTM	ITBM	ω	transfinite
Sym.	OTM	?	transfinite	transfinite

In order to repair this asymmetry one would have to allow BSS registers to store transfinite information and at the same time allow the machine to run for a transfinite amount of time.

The Problem of Limits

As we said, we are interested defining machines which can work with generalisation of the reals. In particular this means that our machines should be able to work with non-archimedean fields.

The Problem of Limits

As we said, we are interested defining machines which can work with generalisation of the reals. In particular this means that our machines should be able to work with non-archimedean fields.

A naive approach to the problem would be that of generalising ITBM machines. Given a non-archimedean Cauchy complete field K define a machine whose registers contain elements of K and that at each step can apply rational functions over K to the registers. For limit stages, as for ITBM our machine would just compute the Cauchy limit of the content of each register at previous stages.

The Problem of Limits

As we said, we are interested defining machines which can work with generalisation of the reals. In particular this means that our machines should be able to work with non-archimedean fields.

A naive approach to the problem would be that of generalising ITBM machines. Given a non-archimedean Cauchy complete field K define a machine whose registers contain elements of K and that at each step can apply rational functions over K to the registers. For limit stages, as for ITBM our machine would just compute the Cauchy limit of the content of each register at previous stages.

Theorem. (Folklore). Let K be a non-archimedean ordered field and s be an infinite sequence of elements of K of length ω. Then, if it is not eventually constant, s diverges.

Overcoming Limits

Given to sets of surreal numbers $L < R$ the Simplicity Theorem gives us a way of choosing a surreal in between them.

Transfinite Computability

Surreal Numbers

Generalised Real Line

Generalised Type 2 Computability

Surreal Blum-Shub-Smale machines

Overcoming Limits

Given to sets of surreal numbers $L < R$ the Simplicity Theorem gives us a way of choosing a surreal in between them.

This gives us a very natural way of defining functions. Many functions over the surreal numbers are indeed defined using the Simplicity Theorem.

Examples are: all the surreal operations, the exponential function, and extensions of analytic functions.

Dedekind Registers

A Dedekind register can be thought as a register containing a surreal which is connected to two stacks L and R and whose content is automatically updated by the machine to $[L|R]$.

- Each Dedekind register has two write only stacks attached L and R;
- The machine automatically updates the content of the register to $[L|R]$ at each step;
- At limit stages the content of each stack is the union of its content at previous stages.

Surreal Blum-Shub-Smale Machines

Given a class of surreal numbers K, a K *surreal Blum-Shub-Smale machine* (SBSS) is a register machine.

- The machine has both normal and Dedekind registers.
- Each register contains a surreal.
- At each (successor) step of the computation the machine can perform a test on a register and jump, or apply a rational function with coefficients in K to some registers.
- At limit Dedekind registers are updated as mentioned before and normal registers whose content is not eventually constant are initialized to 0.

Computable Surreal Functions

Using K-SBSS machines we can define a notion of computability over surreal numbers:

Definition

Let K be a subclass of surreal numbers. A function $F : \mathrm{No} \to \mathrm{No}$ is K-SBSS computable iff there is a K-SBSS machine such that for all $s \in \mathrm{dom}(F)$ stops with output $F(s)$ and diverges for every $s \notin \mathrm{dom}(F)$.

Note that every class K defines a notion of computability.

Computing Sign Sequences

Note that in principle Surreal BSS machines cannot access directly the sign sequence representation of a surreal.

Lemma

Let $\{-1, 0, 1\} \subseteq K$ be a subclass of No. *Then, the following functions are K-SBSS computable:*

1. *The function* $\mathrm{sgn} : \mathsf{No} \times \mathsf{On} \to \{0, 1, 2\}$ *that for every* $\alpha \in \mathsf{On}$ *and* $s \in \mathsf{No}$ *returns* 0 *if the* $1 + \alpha$*th sign in the sign expansion of* s *is* $-$, 1 *if the* $1 + \alpha$*th sign in the sign expansion of* s *is* $+$ *and* 2 *if the sign expansion of* s *is shorter than* $1 + \alpha$*;*
2. *the function* $\mathrm{seg} : \mathsf{No} \times \mathsf{On} \to \mathsf{No}$ *that given a surreal* s *and an ordinal* $\alpha \in \mathrm{dom}(s)$ *returns the surreal whose sign sequence is the initial segment of* s *of length* α.
3. *The function* $\mathrm{cng} : \mathsf{No} \times \mathsf{On} \times \{0, 1\} \to \mathsf{No}$ *that given a surreal* $s \in \mathsf{No}$, $\mathrm{sgn} \in \{0, 1\}$ *and* $\alpha \in \mathsf{On}$ *such that* $\alpha < \mathrm{dom}(s)$ *returns a surreal* $s' \in \mathsf{No}$ *whose sign expansion is obtained by substituting the* $1 + \alpha$*th sign in the expansion of* s *with* $-$ *if* $\mathrm{sgn} = 0$ *and with* $+$ *if* $\mathrm{sgn} = 1$*;*

Surreal Blum-Shub-Smale Machines Power

The fact that SBSS machines can access the sign sequence representation of surreal numbers allows us to simulate tape machines using SBSS machines.

Theorem

Let $\{-1, 0, 1\} \subseteq K$ *be a subclass of* No. *Then, every ITTM-computable function is* K*-SBSS computable.*

Corollary

Let K *be a subclass of* No. *Then, every function computable by an infinite time Blum-Shub-Smale machine is* K*-SBSS computable and the halting problem for infinite time Blum-Shub-Smale machine is* K*-SBSS computable.*

Surreal Blum-Shub-Smale Machines & OTMs

Using techniques similar to those used to simulate ITTMs one can actually prove that SBSS machines are capable to simulate OTMs. On the other hand, using the algorithms we use to prove that surreal operations are OTM computable one can prove that if the coefficients of the rational functions of a programs are OTM computable, then an OTM can simulate the SBSS program.

Theorem

Let $\{-1, 0, 1\} \subseteq K$ *be a subclass of* No*. Then a partial function* $F : \mathrm{No} \to \mathrm{No}$ *is* K*-SBSS computable iff it is computable by an OTM with parameters in* K*.*

Full Surreal Blum-Shub-Smale Machines

Consider the case in which $K = \mathrm{No}$.

Corollary

Every partial function $F : \mathrm{No} \to \mathrm{No}$ *which is a set is* No-*SBSS computable.*

Full Surreal Blum-Shub-Smale Machines

Consider the case in which $K = \mathrm{No}$.

Corollary

Every partial function $F : \mathrm{No} \to \mathrm{No}$ *which is a set is* No-*SBSS computable.*

These machines provide a register model for the *infinite programs machines* (IPM) introduced by Lewis. Whose computational power is that of OTM with parameters in 2^{On}

Theory of Surreal Blum-Shub-Smale Machines

Similarly to the classical theory a Universal Machine Theorem and Halting Problem Theorem are provable for these machines.

Theorem (Universal Machine)

For every class $\{-1, 0, 1\} \subseteq K$ of surreal numbers there is a K-SBSS universal machine.

Theorem (Halting Problem)

For every class $\{-1, 0, 1\} \subseteq K$ of surreal numbers there is a class of surreals which is not K-SBSS computable.

Computing Over RCF

In our joint paper Hugo Nobrega and I used OTMs to define a notion of computability over generalisations of the real line.

Computing Over RCF

In our joint paper Hugo Nobrega and I used OTMs to define a notion of computability over generalisations of the real line.

As we said this construction indirectly shows that surreal operations are OTM computable.

Computing Over RCF

In our joint paper Hugo Nobrega and I used OTMs to define a notion of computability over generalisations of the real line.

As we said this construction indirectly shows that surreal operations are OTM computable.

As they are defined Surreal BSS machines induce a notion of computability over surreal.

Transfinite Computability
Surreal Numbers
Generalised Real Line
Generalised Type 2 Computability
Surreal Blum-Shub-Smale machines

Computing Over RCF

In our joint paper Hugo Nobrega and I used OTMs to define a notion of computability over generalisations of the real line.

As we said this construction indirectly shows that surreal operations are OTM computable.

As they are defined Surreal BSS machines induce a notion of computability over surreal.

Surreal BSS machines induce a notion of transfinite computability over initial subfields of the surreal numbers.

Computing Over RCF

In our joint paper Hugo Nobrega and I used OTMs to define a notion of computability over generalisations of the real line.

As we said this construction indirectly shows that surreal operations are OTM computable.

As they are defined Surreal BSS machines induce a notion of computability over surreal.

Surreal BSS machines induce a notion of transfinite computability over initial subfields of the surreal numbers.

Therefore SBSS machines induce a notion of transfinite computability over every real closed field. In particular over the generalised reals.