Ordinal Computability - Where we are and where we (can) go

Merlin Carl
Universität Konstanz, Fachbereich für Mathematik und Statistik

By the *CTT*, we can get negative results on computability and show that certain problems are not effectively solvable. Such results can be quite relevant for mathematics.

A good example is Hilbert's 10th problem: There is no algorithm for deciding whether an equation $p(x_1, ..., x_n) = 0$ with $p \in \mathbb{Z}[X_1, ..., X_n]$ has an integer solution.

Such results serve at least two purposes: They save us from unsuccesful attempts and they direct our attention towards the achievable.

Recursion Theory in Mathematics

By the *CTT*, we can get negative results on computability and show that certain problems are not effectively solvable. Such results can be quite relevant for mathematics.

A good example is Hilbert's 10th problem: There is no algorithm for deciding whether an equation $p(x_1, ..., x_n) = 0$ with $p \in \mathbb{Z}[X_1, ..., X_n]$ has an integer solution.

Such results serve at least two purposes: They save us from unsuccesful attempts and they direct our attention towards the achievable.

By the *CTT*, we can get negative results on computability and show that certain problems are not effectively solvable. Such results can be quite relevant for mathematics.

A good example is Hilbert's 10th problem: There is no algorithm for deciding whether an equation $p(x_1, ..., x_n) = 0$ with $p \in \mathbb{Z}[X_1, ..., X_n]$ has an integer solution.

Such results serve at least two purposes: They save us from unsuccesful attempts and they direct our attention towards the achievable.

Infinite Procedures in Mathematics

Turing computations model the idea of computing with integers. But the actual (usual) mathematical universe contains objects that cannot be coded by integers. Yet, the concept of mechanical manipulations of such objects is generally accepted as meaningful. Also, in mathematics, we often makes use of infinitary construction methods e.g. in existence proofs and infinitary recursive definitions:

(1) 'Every field has an algebraic closure.'

(2) 'Every integral domain has a field of fractions.'

(3) The definition of the L-hierarchy in set theory.

etc...

'Explicit' proofs of existence are often preferable to indirect proofs, as the construction method can be used as a method of investigating an object by transfinite induction.

This motivates the study of models of infinitary computations that give a precise meaning to various intuitive notions of infinitary constructions.

Infinite Procedures in Mathematics

Turing computations model the idea of computing with integers. But the actual (usual) mathematical universe contains objects that cannot be coded by integers. Yet, the concept of mechanical manipulations of such objects is generally accepted as meaningful. Also, in mathematics, we often makes use of infinitary construction methods e.g. in existence proofs and infinitary recursive definitions:

(1) 'Every field has an algebraic closure.'

(2) 'Every integral domain has a field of fractions.'

(3) The definition of the L-hierarchy in set theory.

etc...

'Explicit' proofs of existence are often preferable to indirect proofs, as the construction method can be used as a method of investigating an object by transfinite induction.

This motivates the study of models of infinitary computations that give a precise meaning to various intuitive notions of infinitary constructions.

Infinite Procedures in Mathematics

Turing computations model the idea of computing with integers. But the actual (usual) mathematical universe contains objects that cannot be coded by integers. Yet, the concept of mechanical manipulations of such objects is generally accepted as meaningful. Also, in mathematics, we often makes use of infinitary construction methods e.g. in existence proofs and infinitary recursive definitions:
(1) 'Every field has an algebraic closure.'
(2) 'Every integral domain has a field of fractions.'
(3) The definition of the L-hierarchy in set theory.
etc...
'Explicit' proofs of existence are often preferable to indirect proofs, as the construction method can be used as a method of investigating an object by transfinite induction.
This motivates the study of models of infinitary computations that give a precise meaning to various intuitive notions of infinitary constructions.

Infinite Procedures in Mathematics

Turing computations model the idea of computing with integers. But the actual (usual) mathematical universe contains objects that cannot be coded by integers. Yet, the concept of mechanical manipulations of such objects is generally accepted as meaningful. Also, in mathematics, we often makes use of infinitary construction methods e.g. in existence proofs and infinitary recursive definitions:
(1) 'Every field has an algebraic closure.'
(2) 'Every integral domain has a field of fractions.'
(3) The definition of the L-hierarchy in set theory.
etc...
'Explicit' proofs of existence are often preferable to indirect proofs, as the construction method can be used as a method of investigating an object by transfinite induction.
This motivates the study of models of infinitary computations that give a precise meaning to various intuitive notions of infinitary constructions.

Effective Mathematics of the Uncountable?

W. Hodges, 'On the Effectivity of some Field Constructions':
'Since the 1930s (Post, Turing) we have known exactly what it is for a function of natural numbers (...) to be *effectively or algorithmically computable*. (...) Now every mathematician is at least vaguely aware of another quite different notion of 'effective function', which has nothing at all to do with denumerable sets.'

(Goes on to offer as examples the function F_0 taking an integral domain to its field of fractions and the function F_1 taking each ring R with identity to a maximal ideal of R.)
'In the sense which concerns us (...), function F_0 is *effective*, function F_1 is (apparently) highly non-effective (...) To prove theorems, we have to replace this intuitive notion of effectiveness with something more precise (...)'

Effective Mathematics of the Uncountable?

W. Hodges, 'On the Effectivity of some Field Constructions':
'Since the 1930s (Post, Turing) we have known exactly what it is for a function of natural numbers (...) to be *effectively or algorithmically computable*. (...) Now every mathematician is at least vaguely aware of another quite different notion of 'effective function', which has nothing at all to do with denumerable sets.'
(Goes on to offer as examples the function F_0 taking an integral domain to its field of fractions and the function F_1 taking each ring R with identity to a maximal ideal of R.)
'In the sense which concerns us (...), function F_0 is *effective*, function F_1 is (apparently) highly non-effective (...) To prove theorems, we have to replace this intuitive notion of effectiveness with something more precise (...)'

W. Hodges, 'On the Effectivity of some Field Constructions':
'Since the 1930s (Post, Turing) we have known exactly what it is for a function of natural numbers (...) to be *effectively or algorithmically computable*. (...) Now every mathematician is at least vaguely aware of another quite different notion of 'effective function', which has nothing at all to do with denumerable sets.'
(Goes on to offer as examples the function F_0 taking an integral domain to its field of fractions and the function F_1 taking each ring R with identity to a maximal ideal of R.)
'In the sense which concerns us (...), function F_0 is *effective*, function F_1 is (apparently) highly non-effective (...) To prove theorems, we have to replace this intuitive notion of effectiveness with something more precise (...)'

Ordinal Turing Machines

(Introduced by P. Koepke in 2005)

OTMs have the same 'software' as Turing machines: Commands that, depending on the current state and the symbol currently read, tell the machine what symbol to write, which new internal state to assume and where to move the read/write head.

Similarly to Turing machines, they have a tape with cells indexed with ordinals (each of which can contain a 0 or a 1), a read/write head, a finite set of internal states, represented by natural numbers and possibly an oracle.

However, the whole class of ordinals is used in the indexing of the tape cells of an OTM and its working time can be an arbitrary ordinal.

Ordinal Turing Machines

(Introduced by P. Koepke in 2005)

OTMs have the same 'software' as Turing machines: Commands that, depending on the current state and the symbol currently read, tell the machine what symbol to write, which new internal state to assume and where to move the read/write head.

Similarly to Turing machines, they have a tape with cells indexed with ordinals (each of which can contain a 0 or a 1), a read/write head, a finite set of internal states, represented by natural numbers and possibly an oracle.

However, the whole class of ordinals is used in the indexing of the tape cells of an OTM and its working time can be an arbitrary ordinal.

Ordinal Turing Machines

(Introduced by P. Koepke in 2005)

OTMs have the same 'software' as Turing machines: Commands that, depending on the current state and the symbol currently read, tell the machine what symbol to write, which new internal state to assume and where to move the read/write head.
Similarly to Turing machines, they have a tape with cells indexed with ordinals (each of which can contain a 0 or a 1), a read/write head, a finite set of internal states, represented by natural numbers and possibly an oracle.
However, the whole class of ordinals is used in the indexing of the tape cells of an OTM and its working time can be an arbitrary ordinal.

Ordinal Turing Machines

(Introduced by P. Koepke in 2005)

OTMs have the same 'software' as Turing machines: Commands that, depending on the current state and the symbol currently read, tell the machine what symbol to write, which new internal state to assume and where to move the read/write head.
Similarly to Turing machines, they have a tape with cells indexed with ordinals (each of which can contain a 0 or a 1), a read/write head, a finite set of internal states, represented by natural numbers and possibly an oracle.
However, the whole class of ordinals is used in the indexing of the tape cells of an OTM and its working time can be an arbitrary ordinal.

Computations along an ordinal time axis

We keep the way a Turing computation works at successor steps. But now, what should the state of the machine be at a limit time λ?

The internal state s_λ at time λ, we set $s_\lambda := \liminf\{s_\iota | \iota < \lambda\}$. The head position p_λ at time λ is $p_\lambda := \liminf\{p_\iota | \iota < \lambda\}$. Note that this limit always exists in the ordinals.

If in an *OTM*-computation the head is moved to the left from a limit ordinal, it is reset to 0.

Concerning the tape content $(t_{\iota\lambda} | \iota \in On)$ at time λ, we set $t_{\iota\lambda} = \liminf\{t_{\iota\gamma} | \gamma < \lambda\}$.

We distinguish two variants: parameter-free *OTM*s start on a tape which contains 0 on every cell with infinite index. A parameter-*OTM* may have also have a single cell with infinite index marked with 1.

Computations along an ordinal time axis

We keep the way a Turing computation works at successor steps.
But now, what should the state of the machine be at a limit time λ?
The internal state s_λ at time λ, we set $s_\lambda := \liminf\{s_\iota | \iota < \lambda\}$.
The head position p_λ at time λ is $p_\lambda := \liminf\{p_\iota | \iota < \lambda\}$. Note that this limit always exists in the ordinals.
If in an *OTM*-computation the head is moved to the left from a limit ordinal, it is reset to 0.
Concerning the tape content $(t_{\iota\lambda} | \iota \in On)$ at time λ, we set $t_{\iota\lambda} = \liminf\{t_{\iota\gamma} | \gamma < \lambda\}$.
We distinguish two variants: parameter-free *OTM*s start on a tape which contains 0 on every cell with infinite index. A parameter-*OTM* may have also have a single cell with infinite index marked with 1.

Computations along an ordinal time axis

We keep the way a Turing computation works at successor steps. But now, what should the state of the machine be at a limit time λ?
The internal state s_λ at time λ, we set $s_\lambda := \liminf\{s_\iota|\iota < \lambda\}$.
The head position p_λ at time λ is $p_\lambda := \liminf\{p_\iota|\iota < \lambda\}$. Note that this limit always exists in the ordinals.
If in an *OTM*-computation the head is moved to the left from a limit ordinal, it is reset to 0.
Concerning the tape content $(t_{\iota\lambda}|\iota \in On)$ at time λ, we set $t_{\iota\lambda} = \liminf\{t_{\iota\gamma}|\gamma < \lambda\}$.
We distinguish two variants: parameter-free *OTM*s start on a tape which contains 0 on every cell with infinite index. A parameter-*OTM* may have also have a single cell with infinite index marked with 1.

Computations along an ordinal time axis

We keep the way a Turing computation works at successor steps. But now, what should the state of the machine be at a limit time λ?

The internal state s_λ at time λ, we set $s_\lambda := \liminf\{s_\iota | \iota < \lambda\}$.

The head position p_λ at time λ is $p_\lambda := \liminf\{p_\iota | \iota < \lambda\}$. Note that this limit always exists in the ordinals.

If in an *OTM*-computation the head is moved to the left from a limit ordinal, it is reset to 0.

Concerning the tape content $(t_{\iota\lambda} | \iota \in On)$ at time λ, we set $t_{\iota\lambda} = \liminf\{t_{\iota\gamma} | \gamma < \lambda\}$.

We distinguish two variants: parameter-free *OTM*s start on a tape which contains 0 on every cell with infinite index. A parameter-*OTM* may have also have a single cell with infinite index marked with 1.

Computations along an ordinal time axis

We keep the way a Turing computation works at successor steps. But now, what should the state of the machine be at a limit time λ?

The internal state s_λ at time λ, we set $s_\lambda := \liminf\{s_\iota | \iota < \lambda\}$.

The head position p_λ at time λ is $p_\lambda := \liminf\{p_\iota | \iota < \lambda\}$. Note that this limit always exists in the ordinals.

If in an *OTM*-computation the head is moved to the left from a limit ordinal, it is reset to 0.

Concerning the tape content $(t_{\iota\lambda} | \iota \in On)$ at time λ, we set $t_{\iota\lambda} = \liminf\{t_{\iota\gamma} | \gamma < \lambda\}$.

We distinguish two variants: parameter-free *OTM*s start on a tape which contains 0 on every cell with infinite index. A parameter-*OTM* may have also have a single cell with infinite index marked with 1.

We keep the way a Turing computation works at successor steps. But now, what should the state of the machine be at a limit time λ?
The internal state s_λ at time λ, we set $s_\lambda := \liminf\{s_\iota | \iota < \lambda\}$.
The head position p_λ at time λ is $p_\lambda := \liminf\{p_\iota | \iota < \lambda\}$. Note that this limit always exists in the ordinals.
If in an *OTM*-computation the head is moved to the left from a limit ordinal, it is reset to 0.
Concerning the tape content $(t_{\iota\lambda} | \iota \in On)$ at time λ, we set $t_{\iota\lambda} = \liminf\{t_{\iota\gamma} | \gamma < \lambda\}$.
We distinguish two variants: parameter-free *OTM*s start on a tape which contains 0 on every cell with infinite index. A parameter-*OTM* may have also have a single cell with infinite index marked with 1.

Infinite time computability

How an *OTM* works should now be clear: Simply run through the program and act according to the commands.
A function $f : \omega \to \omega$ is called *OTM*-computable iff there is a *OTM*-program P that, starting with n on the tape, stops at some ordinal time α with $f(n)$ on the tape.
A subset x of ω is *OTM*-computable if its characteristic function is. As usual, we identify $\mathfrak{P}(\omega)$ with the real numbers.

One obtains a 'zoo' of computational models by restricting the working time and space of OTMs:

- The ITTMs of Hamkins and Kidder - the first model of ordinal computability - have tape length ω.
- The α-TMs of Koepke and Seyfferth have tape length and time bounded by α.
- (α, β)-ITTMs have space bounded by α and time bounded by β.

One obtains a ‘zoo’ of computational models by restricting the working time and space of OTMs:

- The ITTMs of Hamkins and Kidder - the first model of ordinal computability - have tape length ω.
- The α-TMs of Koepke and Seyfferth have tape length and time bounded by α.
- (α, β)-ITTMs have space bounded by α and time bounded by β.

One obtains a ‘zoo’ of computational models by restricting the working time and space of OTMs:

- The ITTMs of Hamkins and Kidder - the first model of ordinal computability - have tape length ω.
- The α-TMs of Koepke and Seyfferth have tape length and time bounded by α.
- (α, β)-ITTMs have space bounded by α and time bounded by β.

One obtains a ‘zoo’ of computational models by restricting the working time and space of OTMs:

- The ITTMs of Hamkins and Kidder - the first model of ordinal computability - have tape length ω.
- The α-TMs of Koepke and Seyfferth have tape length and time bounded by α.
- (α, β)-ITTMs have space bounded by α and time bounded by β.

Similarly, one can generalize register machines to the transfinite:

(Koepke) An Ordinal Register Machine (ORM) has finitely many registers, each of which can store a single ordinal. ORM-programs are regular register machine programs: finitely many numbered program lines, commands for incrementing a register content, resetting a register content back to 0, copying the content of one register to another, and a conditional jump to a certain program line that takes place when some register contains 0.

At limit times, the active program line is the liminf of the sequence of earlier program lines, and the register contents are also obtained as liminfs of the sequences of earlier contents.

Similarly, one can generalize register machines to the transfinite:

(Koepke) An Ordinal Register Machine (ORM) has finitely many registers, each of which can store a single ordinal. ORM-programs are regular register machine programs: finitely many numbered program lines, commands for incrementing a register content, resetting a register content back to 0, copying the content of one register to another, and a conditional jump to a certain program line that takes place when some register contains 0.

At limit times, the active program line is the liminf of the sequence of earlier program lines, and the register contents are also obtained as liminfs of the sequences of earlier contents.

Similarly, one can generalize register machines to the transfinite:

(Koepke) An Ordinal Register Machine (ORM) has finitely many registers, each of which can store a single ordinal. ORM-programs are regular register machine programs: finitely many numbered program lines, commands for incrementing a register content, resetting a register content back to 0, copying the content of one register to another, and a conditional jump to a certain program line that takes place when some register contains 0.

At limit times, the active program line is the liminf of the sequence of earlier program lines, and the register contents are also obtained as liminfs of the sequences of earlier contents.

Similarly, one can generalize register machines to the transfinite:

(Koepke) An Ordinal Register Machine (ORM) has finitely many registers, each of which can store a single ordinal. ORM-programs are regular register machine programs: finitely many numbered program lines, commands for incrementing a register content, resetting a register content back to 0, copying the content of one register to another, and a conditional jump to a certain program line that takes place when some register contains 0.

At limit times, the active program line is the liminf of the sequence of earlier program lines, and the register contents are also obtained as liminfs of the sequences of earlier contents.

Again, one can obtain a family of weakenings by restricting the working time and the register contents, due to Koepke.

When restricting the register content to ordinals $< \beta$, one must deal with the possibility of an 'overflow', i.e. the possibility that the program determines that a register should have a content $\geq \beta$ at some point.

(1) The 'weak' variant is that in such a case, the configuration is undefined and the computations 'crashes'.

(2) The 'strong' variant is that the respective registers are set back to 0. This is our default option.

Again, one can obtain a family of weakenings by restricting the working time and the register contents, due to Koepke.
When restricting the register content to ordinals $< \beta$, one must deal with the possibility of an 'overflow', i.e. the possibility that the program determines that a register should have a content $\geq \beta$ at some point.

(1) The 'weak' variant is that in such a case, the configuration is undefined and the computations 'crashes'.

(2) The 'strong' variant is that the respective registers are set back to 0. This is our default option.

Again, one can obtain a family of weakenings by restricting the working time and the register contents, due to Koepke.
When restricting the register content to ordinals $< \beta$, one must deal with the possibility of an 'overflow', i.e. the possibility that the program determines that a register should have a content $\geq \beta$ at some point.

(1) The 'weak' variant is that in such a case, the configuration is undefined and the computations 'crashes'.

(2) The 'strong' variant is that the respective registers are set back to 0. This is our default option.

Again, one can obtain a family of weakenings by restricting the working time and the register contents, due to Koepke.
When restricting the register content to ordinals $< \beta$, one must deal with the possibility of an 'overflow', i.e. the possibility that the program determines that a register should have a content $\geq \beta$ at some point.

(1) The 'weak' variant is that in such a case, the configuration is undefined and the computations 'crashes'.

(2) The 'strong' variant is that the respective registers are set back to 0. This is our default option.

- wITRMs are weak ORMs with registers restricted to ω.
- ITRMs are ORMs with registers restricted to ω.
- α-RMs are ORMs with time and registers restricted to α. (There is no distinction between 'strong' and 'weak' here.)
- α-wITRMs are weak ORMs with registers restricted to α. Similarly: (α, β)-wITRMs.
- α-ITRMs are ORMs with registers restricted to α. Similarly: (α, β)-ITRMs.

In most cases, we know variants of the tape models equivalent to these models (e.g. by imposing on ITTMs that only finitely many cells may contain a 1 at each time, one obtains a model equivalent to wITRMs). It would be nice to complete the picture.

- wITRMs are weak ORMs with registers restricted to ω.
- ITRMs are ORMs with registers restricted to ω.
- α-RMs are ORMs with time and registers restricted to α. (There is no distinction between 'strong' and 'weak' here.)
- α-wITRMs are weak ORMs with registers restricted to α. Similarly: (α, β)-wITRMs.
- α-ITRMs are ORMs with registers restricted to α. Similarly: (α, β)-ITRMs.

In most cases, we know variants of the tape models equivalent to these models (e.g. by imposing on ITTMs that only finitely many cells may contain a 1 at each time, one obtains a model equivalent to wITRMs). It would be nice to complete the picture.

- wITRMs are weak ORMs with registers restricted to ω.
- ITRMs are ORMs with registers restricted to ω.
- α-RMs are ORMs with time and registers restricted to α. (There is no distinction between 'strong' and 'weak' here.)
- α-wITRMs are weak ORMs with registers restricted to α. Similarly: (α, β)-wITRMs.
- α-ITRMs are ORMs with registers restricted to α. Similarly: (α, β)-ITRMs.

In most cases, we know variants of the tape models equivalent to these models (e.g. by imposing on ITTMs that only finitely many cells may contain a 1 at each time, one obtains a model equivalent to wITRMs). It would be nice to complete the picture.

- wITRMs are weak ORMs with registers restricted to ω.
- ITRMs are ORMs with registers restricted to ω.
- α-RMs are ORMs with time and registers restricted to α. (There is no distinction between 'strong' and 'weak' here.)
- α-wITRMs are weak ORMs with registers restricted to α. Similarly: (α, β)-wITRMs.
- α-ITRMs are ORMs with registers restricted to α. Similarly: (α, β)-ITRMs.

In most cases, we know variants of the tape models equivalent to these models (e.g. by imposing on ITTMs that only finitely many cells may contain a 1 at each time, one obtains a model equivalent to wITRMs). It would be nice to complete the picture.

- wITRMs are weak ORMs with registers restricted to ω.
- ITRMs are ORMs with registers restricted to ω.
- α-RMs are ORMs with time and registers restricted to α. (There is no distinction between 'strong' and 'weak' here.)
- α-wITRMs are weak ORMs with registers restricted to α. Similarly: (α, β)-wITRMs.
- α-ITRMs are ORMs with registers restricted to α. Similarly: (α, β)-ITRMs.

In most cases, we know variants of the tape models equivalent to these models (e.g. by imposing on ITTMs that only finitely many cells may contain a 1 at each time, one obtains a model equivalent to wITRMs). It would be nice to complete the picture.

- wITRMs are weak ORMs with registers restricted to ω.
- ITRMs are ORMs with registers restricted to ω.
- α-RMs are ORMs with time and registers restricted to α. (There is no distinction between 'strong' and 'weak' here.)
- α-wITRMs are weak ORMs with registers restricted to α. Similarly: (α, β)-wITRMs.
- α-ITRMs are ORMs with registers restricted to α. Similarly: (α, β)-ITRMs.

In most cases, we know variants of the tape models equivalent to these models (e.g. by imposing on ITTMs that only finitely many cells may contain a 1 at each time, one obtains a model equivalent to wITRMs). It would be nice to complete the picture.

2: COMPUTABILITY

The tape models

	Tape length ω	Tape length α multiplicatively closed	Tape length On
Time ω	$\Delta_1(L_\omega) = \Delta^0_1$ (Folklore)	$\Delta_1(L_\omega) = \Delta^0_1$ (Folklore)	$\Delta_1(L_\omega) = \Delta^0_1$ (Folklore)
Time β admissible	$\mathfrak{P}(\omega) \cap L_{\mathrm{gap}_\alpha(\beta)}$	$\Delta_1(L_{\min(\alpha,\beta)})$ if $\beta \leq \alpha$ and α is exponentially closed [1] $L_{\mathrm{gap}_\alpha(\beta)} \cap \mathfrak{P}(\alpha)$ if $\beta > \alpha$	$\Delta_1(L_\beta)$ [1]
Time On	$\mathfrak{P}(\omega) \cap L_\lambda$ [2]	$\mathfrak{P}(\alpha) \cap L_{\lambda(\alpha)}$	$\mathfrak{P}(\mathrm{On}) \cap L$ [3]

[1] Koepke and Seyfferth
[2] Hamkins, Lewis and Welch
[3] Koepke

Register Models

The following results expect the last one are classical:

- (Koepke) $x \subseteq \omega$ is wITRM-computable if and only if $x \in L_{\omega_1^{CK}}$.
- (Koepke) $x \subseteq \omega$ is ITRM-computable if and only if $x \in L_{\omega_\omega^{CK}}$.
- (Koepke) $x \subseteq \mathrm{On}$ is ORM-computable with parameters if and only if $x \in L$.
- (Schlicht, Seyfferth, C.) $x \subseteq \omega$ is ORM-computable without parameters if and only if $x \in L_\sigma$, where σ is minimal such that $L_\sigma \prec_{\Sigma_1} L$.
- (Adaptation of the Koepke-Seyfferth argument for α-TMs) For exponentially closed α, $x \subseteq \alpha$ is α-RM-computable with/without parameters if and only if x is $\Delta_1(L_\alpha)/\mathbf{\Delta}_1(L_\alpha)$.
- (C.) For $k \in \omega$, the (ω, ω_k^{CK})-ITRM-computable real numbers are exactly those in $L_{\omega_k^{CK}}$.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$

Register Models

The following results expect the last one are classical:

- (Koepke) $x \subseteq \omega$ is wITRM-computable if and only if $x \in L_{\omega_1^{CK}}$.
- (Koepke) $x \subseteq \omega$ is ITRM-computable if and only if $x \in L_{\omega_\omega^{CK}}$.
- (Koepke) $x \subseteq$ On is ORM-computable with parameters if and only if $x \in L$.
- (Schlicht, Seyfferth, C.) $x \subseteq \omega$ is ORM-computable without parameters if and only if $x \in L_\sigma$, where σ is minimal such that $L_\sigma \prec_{\Sigma_1} L$.
- (Adaptation of the Koepke-Seyfferth argument for α-TMs) For exponentially closed α, $x \subseteq \alpha$ is α-RM-computable with/without parameters if and only if x is $\Delta_1(L_\alpha)/\boldsymbol{\Delta}_1(L_\alpha)$.
- (C.) For $k \in \omega$, the (ω, ω_k^{CK})-ITRM-computable real numbers are exactly those in $L_{\omega_k^{CK}}$.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$.

Register Models

The following results expect the last one are classical:

- (Koepke) $x \subseteq \omega$ is wITRM-computable if and only if $x \in L_{\omega_1^{CK}}$.
- (Koepke) $x \subseteq \omega$ is ITRM-computable if and only if $x \in L_{\omega_\omega^{CK}}$.
- (Koepke) $x \subseteq$ On is ORM-computable with parameters if and only if $x \in L$.
- (Schlicht, Seyfferth, C.) $x \subseteq \omega$ is ORM-computable without parameters if and only if $x \in L_\sigma$, where σ is minimal such that $L_\sigma \prec_{\Sigma_1} L$.
- (Adaptation of the Koepke-Seyfferth argument for α-TMs) For exponentially closed α, $x \subseteq \alpha$ is α-RM-computable with/without parameters if and only if x is $\Delta_1(L_\alpha)/\mathbf{\Delta}_1(L_\alpha)$.
- (C.) For $k \in \omega$, the (ω, ω_k^{CK})-ITRM-computable real numbers are exactly those in $L_{\omega_k^{CK}}$.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$

Register Models

The following results expect the last one are classical:

- (Koepke) $x \subseteq \omega$ is wITRM-computable if and only if $x \in L_{\omega_1^{CK}}$.
- (Koepke) $x \subseteq \omega$ is ITRM-computable if and only if $x \in L_{\omega_\omega^{CK}}$.
- (Koepke) $x \subseteq \mathrm{On}$ is ORM-computable with parameters if and only if $x \in L$.
- (Schlicht, Seyfferth, C.) $x \subseteq \omega$ is ORM-computable without parameters if and only if $x \in L_\sigma$, where σ is minimal such that $L_\sigma \prec_{\Sigma_1} L$.
- (Adaptation of the Koepke-Seyfferth argument for α-TMs) For exponentially closed α, $x \subseteq \alpha$ is α-RM-computable with/without parameters if and only if x is $\Delta_1(L_\alpha)/\mathbf{\Delta}_1(L_\alpha)$.
- (C.) For $k \in \omega$, the (ω, ω_k^{CK})-ITRM-computable real numbers are exactly those in $L_{\omega_k^{CK}}$.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$

Register Models

The following results expect the last one are classical:

- (Koepke) $x \subseteq \omega$ is wITRM-computable if and only if $x \in L_{\omega_1^{\mathrm{CK}}}$.
- (Koepke) $x \subseteq \omega$ is ITRM-computable if and only if $x \in L_{\omega_\omega^{\mathrm{CK}}}$.
- (Koepke) $x \subseteq \mathrm{On}$ is ORM-computable with parameters if and only if $x \in L$.
- (Schlicht, Seyfferth, C.) $x \subseteq \omega$ is ORM-computable without parameters if and only if $x \in L_\sigma$, where σ is minimal such that $L_\sigma \prec_{\Sigma_1} L$.
- (Adaptation of the Koepke-Seyfferth argument for α-TMs) For exponentially closed α, $x \subseteq \alpha$ is α-RM-computable with/without parameters if and only if x is $\Delta_1(L_\alpha)/\boldsymbol{\Delta}_1(L_\alpha)$.
- (C.) For $k \in \omega$, the $(\omega, \omega_k^{\mathrm{CK}})$-ITRM-computable real numbers are exactly those in $L_{\omega_k^{\mathrm{CK}}}$.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$

Register Models

The following results expect the last one are classical:

- (Koepke) $x \subseteq \omega$ is wITRM-computable if and only if $x \in L_{\omega_1^{\mathrm{CK}}}$.
- (Koepke) $x \subseteq \omega$ is ITRM-computable if and only if $x \in L_{\omega_\omega^{\mathrm{CK}}}$.
- (Koepke) $x \subseteq \mathrm{On}$ is ORM-computable with parameters if and only if $x \in L$.
- (Schlicht, Seyfferth, C.) $x \subseteq \omega$ is ORM-computable without parameters if and only if $x \in L_\sigma$, where σ is minimal such that $L_\sigma \prec_{\Sigma_1} L$.
- (Adaptation of the Koepke-Seyfferth argument for α-TMs) For exponentially closed α, $x \subseteq \alpha$ is α-RM-computable with/without parameters if and only if x is $\Delta_1(L_\alpha)/\boldsymbol{\Delta}_1(L_\alpha)$.
- (C.) For $k \in \omega$, the $(\omega, \omega_k^{\mathrm{CK}})$-ITRM-computable real numbers are exactly those in $L_{\omega_k^{\mathrm{CK}}}$.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$

Register Models

The following results expect the last one are classical:

- (Koepke) $x \subseteq \omega$ is wITRM-computable if and only if $x \in L_{\omega_1^{\mathrm{CK}}}$.
- (Koepke) $x \subseteq \omega$ is ITRM-computable if and only if $x \in L_{\omega_\omega^{\mathrm{CK}}}$.
- (Koepke) $x \subseteq \mathrm{On}$ is ORM-computable with parameters if and only if $x \in L$.
- (Schlicht, Seyfferth, C.) $x \subseteq \omega$ is ORM-computable without parameters if and only if $x \in L_\sigma$, where σ is minimal such that $L_\sigma \prec_{\Sigma_1} L$.
- (Adaptation of the Koepke-Seyfferth argument for α-TMs) For exponentially closed α, $x \subseteq \alpha$ is α-RM-computable with/without parameters if and only if x is $\Delta_1(L_\alpha)/\boldsymbol{\Delta}_1(L_\alpha)$.
- (C.) For $k \in \omega$, the $(\omega, \omega_k^{\mathrm{CK}})$-ITRM-computable real numbers are exactly those in $L_{\omega_k^{\mathrm{CK}}}$.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$.
Both is quite false:

- (C.) If α is Σ_2-admissible, then α-wITRM-computability coincides with α-RM-computability, i.e. with $\mathbf{\Delta}_1(L_\alpha)$. Thus, the 'extra time' beyond α does not help.
- (C.) On the other hand, ω_1^{CK}-wITRMs compute at least up to $L_{\omega_\omega^{CK}}$, and possibly further (we don't know).
- (C., Schlicht) If $\alpha > \omega$ is regular in L_{α^ω} (in particular, when α is a regular uncountable cardinal), then the α-ITRM-computable subsets of α are exactly those in ... $L_{\alpha+1}$.

Thus, α-ITRMs and α-wITRMs are apparently 'wild' with respect to computational strength. We are far from a complete understanding here, even less for (α,β)-(w)ITRMs. However, the results above indicate where one might look for further instructive special cases: A further analysis of ω_1^{CK}-(w)ITRMs should both be feasible with the given methods and reveal something about the general problem.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$.
Both is quite false:

- (C.) If α is Σ_2-admissible, then α-wITRM-computability coincides with α-RM-computability, i.e. with $\mathbf{\Delta}_1(L_\alpha)$. Thus, the 'extra time' beyond α does not help.
- (C.) On the other hand, ω_1^{CK}-wITRMs compute at least up to $L_{\omega_\omega^{\text{CK}}}$, and possibly further (we don't know).
- (C., Schlicht) If $\alpha > \omega$ is regular in L_{α^ω} (in particular, when α is a regular uncountable cardinal), then the α-ITRM-computable subsets of α are exactly those in ... $L_{\alpha+1}$.

Thus, α-ITRMs and α-wITRMs are apparently 'wild' with respect to computational strength. We are far from a complete understanding here, even less for (α, β)-(w)ITRMs. However, the results above indicate where one might look for further instructive special cases: A further analysis of ω_1^{CK}-(w)ITRMs should both be feasible with the given methods and reveal something about the general problem.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$.
Both is quite false:

- (C.) If α is Σ_2-admissible, then α-wITRM-computability coincides with α-RM-computability, i.e. with $\mathbf{\Delta}_1(L_\alpha)$. Thus, the 'extra time' beyond α does not help.
- (C.) On the other hand, ω_1^{CK}-wITRMs compute at least up to $L_{\omega_\omega^{\mathrm{CK}}}$, and possibly further (we don't know).
- (C., Schlicht) If $\alpha > \omega$ is regular in L_{α^ω} (in particular, when α is a regular uncountable cardinal), then the α-ITRM-computable subsets of α are exactly those in ... $L_{\alpha+1}$.

Thus, α-ITRMs and α-wITRMs are apparently 'wild' with respect to computational strength. We are far from a complete understanding here, even less for (α, β)-(w)ITRMs. However, the results above indicate where one might look for further instructive special cases: A further analysis of ω_1^{CK}-(w)ITRMs should both be feasible with the given methods and reveal something about the general problem.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$.
Both is quite false:

- (C.) If α is Σ_2-admissible, then α-wITRM-computability coincides with α-RM-computability, i.e. with $\mathbf{\Delta}_1(L_\alpha)$. Thus, the 'extra time' beyond α does not help.
- (C.) On the other hand, ω_1^{CK}-wITRMs compute at least up to $L_{\omega_\omega^{\mathrm{CK}}}$, and possibly further (we don't know).
- (C., Schlicht) If $\alpha > \omega$ is regular in L_{α^ω} (in particular, when α is a regular uncountable cardinal), then the α-ITRM-computable subsets of α are exactly those in ... $L_{\alpha+1}$.

Thus, α-ITRMs and α-wITRMs are apparently 'wild' with respect to computational strength. We are far from a complete understanding here, even less for (α, β)-(w)ITRMs. However, the results above indicate where one might look for further instructive special cases: A further analysis of ω_1^{CK}-(w)ITRMs should both be feasible with the given methods and reveal something about the general problem.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$.
Both is quite false:

- (C.) If α is Σ_2-admissible, then α-wITRM-computability coincides with α-RM-computability, i.e. with $\mathbf{\Delta}_1(L_\alpha)$. Thus, the 'extra time' beyond α does not help.
- (C.) On the other hand, ω_1^{CK}-wITRMs compute at least up to $L_{\omega_\omega^{\mathrm{CK}}}$, and possibly further (we don't know).
- (C., Schlicht) If $\alpha > \omega$ is regular in L_{α^ω} (in particular, when α is a regular uncountable cardinal), then the α-ITRM-computable subsets of α are exactly those in ... $L_{\alpha+1}$.

Thus, α-ITRMs and α-wITRMs are apparently 'wild' with respect to computational strength. We are far from a complete understanding here, even less for (α, β)-(w)ITRMs. However, the results above indicate where one might look for further instructive special cases: A further analysis of ω_1^{CK}-(w)ITRMs should both be feasible with the given methods and reveal something about the general problem.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$.
Both is quite false:

- (C.) If α is Σ_2-admissible, then α-wITRM-computability coincides with α-RM-computability, i.e. with $\mathbf{\Delta}_1(L_\alpha)$. Thus, the 'extra time' beyond α does not help.
- (C.) On the other hand, ω_1^{CK}-wITRMs compute at least up to $L_{\omega_\omega^{\mathrm{CK}}}$, and possibly further (we don't know).
- (C., Schlicht) If $\alpha > \omega$ is regular in L_{α^ω} (in particular, when α is a regular uncountable cardinal), then the α-ITRM-computable subsets of α are exactly those in ... $L_{\alpha+1}$.

Thus, α-ITRMs and α-wITRMs are apparently 'wild' with respect to computational strength. We are far from a complete understanding here, even less for (α, β)-(w)ITRMs. However, the results above indicate where one might look for further instructive special cases: A further analysis of ω_1^{CK}-(w)ITRMs should both be feasible with the given methods and reveal something about the general problem.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$.
Both is quite false:

- (C.) If α is Σ_2-admissible, then α-wITRM-computability coincides with α-RM-computability, i.e. with $\mathbf{\Delta}_1(L_\alpha)$. Thus, the 'extra time' beyond α does not help.
- (C.) On the other hand, ω_1^{CK}-wITRMs compute at least up to $L_{\omega_\omega^{\mathrm{CK}}}$, and possibly further (we don't know).
- (C., Schlicht) If $\alpha > \omega$ is regular in L_{α^ω} (in particular, when α is a regular uncountable cardinal), then the α-ITRM-computable subsets of α are exactly those in ... $L_{\alpha+1}$.

Thus, α-ITRMs and α-wITRMs are apparently 'wild' with respect to computational strength. We are far from a complete understanding here, even less for (α, β)-(w)ITRMs. However, the results above indicate where one might look for further instructive special cases: A further analysis of ω_1^{CK}-(w)ITRMs should both be feasible with the given methods and reveal something about the general problem.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$.
Both is quite false:

- (C.) If α is Σ_2-admissible, then α-wITRM-computability coincides with α-RM-computability, i.e. with $\mathbf{\Delta}_1(L_\alpha)$. Thus, the 'extra time' beyond α does not help.
- (C.) On the other hand, ω_1^{CK}-wITRMs compute at least up to $L_{\omega_\omega^{\mathrm{CK}}}$, and possibly further (we don't know).
- (C., Schlicht) If $\alpha > \omega$ is regular in L_{α^ω} (in particular, when α is a regular uncountable cardinal), then the α-ITRM-computable subsets of α are exactly those in ... $L_{\alpha+1}$.

Thus, α-ITRMs and α-wITRMs are apparently 'wild' with respect to computational strength. We are far from a complete understanding here, even less for (α, β)-(w)ITRMs. However, the results above indicate where one might look for further instructive special cases: A further analysis of ω_1^{CK}-(w)ITRMs should both be feasible with the given methods and reveal something about the general problem.

In particular based on the first two, one might conjecture: α-wITRMs compute up to L_{α^+}, α-ITRMs compute up to $L_{\alpha^{+\omega}}$.
Both is quite false:

- (C.) If α is Σ_2-admissible, then α-wITRM-computability coincides with α-RM-computability, i.e. with $\mathbf{\Delta}_1(L_\alpha)$. Thus, the 'extra time' beyond α does not help.
- (C.) On the other hand, ω_1^{CK}-wITRMs compute at least up to $L_{\omega_\omega^{\mathrm{CK}}}$, and possibly further (we don't know).
- (C., Schlicht) If $\alpha > \omega$ is regular in L_{α^ω} (in particular, when α is a regular uncountable cardinal), then the α-ITRM-computable subsets of α are exactly those in ... $L_{\alpha+1}$.

Thus, α-ITRMs and α-wITRMs are apparently 'wild' with respect to computational strength. We are far from a complete understanding here, even less for (α, β)-(w)ITRMs. However, the results above indicate where one might look for further instructive special cases: A further analysis of ω_1^{CK}-(w)ITRMs should both be feasible with the given methods and reveal something about the general problem.

Other Models

- Infinite Time Blum-Shub-Smale Machines (Koepke and Seyfferth)
- Hypermachines (Friedman and Welch)
- Ordinal λ-calculus (Fischbach and Seyfferth)
- Ordinal Kripke equational calculus (Fischbach)
- Deterministic Ordinal Automata (C.)
- ...

Other Models

- Infinite Time Blum-Shub-Smale Machines (Koepke and Seyfferth)
- Hypermachines (Friedman and Welch)
- Ordinal λ-calculus (Fischbach and Seyfferth)
- Ordinal Kripke equational calculus (Fischbach)
- Deterministic Ordinal Automata (C.)
- ...

Other Models

- Infinite Time Blum-Shub-Smale Machines (Koepke and Seyfferth)
- Hypermachines (Friedman and Welch)
- Ordinal λ-calculus (Fischbach and Seyfferth)
- Ordinal Kripke equational calculus (Fischbach)
- Deterministic Ordinal Automata (C.)
- ...

Other Models

- Infinite Time Blum-Shub-Smale Machines (Koepke and Seyfferth)
- Hypermachines (Friedman and Welch)
- Ordinal λ-calculus (Fischbach and Seyfferth)
- Ordinal Kripke equational calculus (Fischbach)
- Deterministic Ordinal Automata (C.)
- ...

Other Models

- Infinite Time Blum-Shub-Smale Machines (Koepke and Seyfferth)
- Hypermachines (Friedman and Welch)
- Ordinal λ-calculus (Fischbach and Seyfferth)
- Ordinal Kripke equational calculus (Fischbach)
- Deterministic Ordinal Automata (C.)
- ...

Other Models

- Infinite Time Blum-Shub-Smale Machines (Koepke and Seyfferth)
- Hypermachines (Friedman and Welch)
- Ordinal λ-calculus (Fischbach and Seyfferth)
- Ordinal Kripke equational calculus (Fischbach)
- Deterministic Ordinal Automata (C.)
- ...

3: Recognizability

4: RECOGNIZABILITY (joint work with Philipp Schlicht and Philip Welch)

From a set-theoretical point of view, infinitary computability is often seen as ‘restrictive’, because many objects of set-theoretical interest - like most types of large cardinals - fall outside of the realm of the computable for the models considered so far.

In fact, there appears to be a ‘constructible barrier’: Every object approachable with infinitary computability seems to belong to the minimal transitive class model of ZFC, namely L.

There have thus been repeated demands to extend infinitary computability beyond that barrier. Our work is an approach to achieving this.

From a set-theoretical point of view, infinitary computability is often seen as ‘restrictive’, because many objects of set-theoretical interest - like most types of large cardinals - fall outside of the realm of the computable for the models considered so far.

In fact, there appears to be a ‘constructible barrier’: Every object approachable with infinitary computability seems to belong to the minimal transitive class model of ZFC, namely L.

There have thus been repeated demands to extend infinitary computability beyond that barrier. Our work is an approach to achieving this.

From a set-theoretical point of view, infinitary computability is often seen as ‘restrictive’, because many objects of set-theoretical interest - like most types of large cardinals - fall outside of the realm of the computable for the models considered so far.

In fact, there appears to be a ‘constructible barrier’: Every object approachable with infinitary computability seems to belong to the minimal transitive class model of ZFC, namely L.

There have thus been repeated demands to extend infinitary computability beyond that barrier. Our work is an approach to achieving this.

A notion related to computability is recognizability:

A set X of ordinals is OTM-**recognizable** if and only if there is an OTM-program P such that, for all sets Y of ordinals, we have $P^Y \downarrow = \delta(X, Y)$ (where δ is the Kronecker symbol).

Thus, P can recognize whether the 'object it observes is X'. Of course, the concept makes sense for any machine type. It was originally defined by Hamkins for ITTMs.

Denote by REC_α the set of real numbers recognizable in the parameter α, let $\text{REC}_\infty := \bigcup_{\iota \in \text{On}} \text{REC}_\iota$. Also, abbreviate $\text{REC}_\emptyset$ by REC.

A notion related to computability is recognizability:

A set X of ordinals is OTM-**recognizable** if and only if there is an OTM-program P such that, for all sets Y of ordinals, we have $P^Y \downarrow = \delta(X, Y)$ (where δ is the Kronecker symbol).

Thus, P can recognize whether the 'object it observes is X'. Of course, the concept makes sense for any machine type. It was originally defined by Hamkins for ITTMs.

Denote by REC_α the set of real numbers recognizable in the parameter α, let $\text{REC}_\infty := \bigcup_{\iota \in \text{On}} \text{REC}_\iota$. Also, abbreviate $\text{REC}_\emptyset$ by REC.

A notion related to computability is recognizability:

A set X of ordinals is OTM-**recognizable** if and only if there is an OTM-program P such that, for all sets Y of ordinals, we have $P^Y\downarrow=\delta(X,Y)$ (where δ is the Kronecker symbol).

Thus, P can recognize whether the 'object it observes is X'. Of course, the concept makes sense for any machine type. It was originally defined by Hamkins for ITTMs.

Denote by REC_α the set of real numbers recognizable in the parameter α, let $\text{REC}_\infty:=\bigcup_{\iota\in\text{On}}\text{REC}_\iota$. Also, abbreviate $\text{REC}_\emptyset$ by REC.

A notion related to computability is recognizability:

A set X of ordinals is OTM-**recognizable** if and only if there is an OTM-program P such that, for all sets Y of ordinals, we have $P^Y\downarrow=\delta(X,Y)$ (where δ is the Kronecker symbol).

Thus, P can recognize whether the 'object it observes is X'. Of course, the concept makes sense for any machine type. It was originally defined by Hamkins for ITTMs.

Denote by REC_α the set of real numbers recognizable in the parameter α, let $\text{REC}_\infty := \bigcup_{\iota\in\text{On}}\text{REC}_\iota$. Also, abbreviate $\text{REC}_\emptyset$ by REC.

Parameter-free recognizable reals are constructible

Fact: REC$\subset L$.

Proof idea: Suppose $x \in RECOG$, and let P be an OTM-program recognizing x.

The statement ψ:='There is $y \subseteq \omega$ such that $P^y \downarrow= 1$' is Σ^1_2 and holds in V by assumption.

By Shoenfield's absoluteness theorem, Σ^1_2-statements are absolute between transitive models of ZFC.

Hence, ψ holds in L. So L contains a real z such that $P^z \downarrow= 1$. As P recognizes x, we must have $x = z$, so $x \in L$.

In general, however, recognizability is strictly weaker than computability.

Fact: REC$\subset$ L.

Proof idea: Suppose $x \in RECOG$, and let P be an OTM-program recognizing x.

The statement ψ:='There is $y \subseteq \omega$ such that $P^y \downarrow = 1$' is Σ^1_2 and holds in V by assumption.

By Shoenfield's absoluteness theorem, Σ^1_2-statements are absolute between transitive models of ZFC.

Hence, ψ holds in L. So L contains a real z such that $P^z \downarrow = 1$. As P recognizes x, we must have $x = z$, so $x \in L$.

In general, however, recognizability is strictly weaker than computability.

What is the connection between recognizability and computability?

Obviously, every computable real r is recognizable: Just compute each bit and compare it with the oracle.
What about the converse?
We will consider this below, both for OTMs with and without parameters.
A recognizable, but non-computable real is called a 'Lost Melody' (Hamkins). There are lost melodies for ITTMs (Hamkins) and ITRMs (C., Koepke), but not for wITRMs (C.) or OTMs without parameters (Dawson, C. (independently), while this question is independent of ZFC for OTMs with parameters.

What is the connection between recognizability and computability?

Obviously, every computable real r is recognizable: Just compute each bit and compare it with the oracle.
What about the converse?
We will consider this below, both for OTMs with and without parameters.
A recognizable, but non-computable real is called a 'Lost Melody' (Hamkins). There are lost melodies for ITTMs (Hamkins) and ITRMs (C., Koepke), but not for wITRMs (C.) or OTMs without parameters (Dawson, C. (independently), while this question is independent of ZFC for OTMs with parameters.

What is the connection between recognizability and computability?

Obviously, every computable real r is recognizable: Just compute each bit and compare it with the oracle.
What about the converse?
We will consider this below, both for OTMs with and without parameters.
A recognizable, but non-computable real is called a 'Lost Melody' (Hamkins). There are lost melodies for ITTMs (Hamkins) and ITRMs (C., Koepke), but not for wITRMs (C.) or OTMs without parameters (Dawson, C. (independently), while this question is independent of ZFC for OTMs with parameters.

Like computability, recognizability can be relativized:
Let $X, Y \subseteq \alpha \in \text{On}$. Then X is recognizable from Y in the parameter $\beta \in \text{On}$ if and only if there is an OTM-program P such that, for all $Z \subseteq \alpha$, we have $P^{Z \oplus Y}(\beta) \downarrow = \delta(Z, X)$.

Intuitively, Y helps in recognizing X.
We only consider real numbers (i.e. subsets of ω) in this talk from now on.

We denote the closure of $\emptyset$ under relative recognizability without parameters by $\mathcal{C}$, with parameter α by $\mathcal{C}_\alpha$ and with arbitrary parameters by $\mathcal{C}^\infty$.

Like computability, recognizability can be relativized:
Let $X, Y \subseteq \alpha \in \text{On}$. Then X is recognizable from Y in the parameter $\beta \in \text{On}$ if and only if there is an OTM-program P such that, for all $Z \subseteq \alpha$, we have $P^{Z \oplus Y}(\beta) \downarrow = \delta(Z, X)$.

Intuitively, Y helps in recognizing X.
We only consider real numbers (i.e. subsets of ω) in this talk from now on.

We denote the closure of $\emptyset$ under relative recognizability without parameters by $\mathcal{C}$, with parameter α by $\mathcal{C}_\alpha$ and with arbitrary parameters by $\mathcal{C}^\infty$.

Like computability, recognizability can be relativized:
Let $X, Y \subseteq \alpha \in \mathrm{On}$. Then X is recognizable from Y in the parameter $\beta \in \mathrm{On}$ if and only if there is an OTM-program P such that, for all $Z \subseteq \alpha$, we have $P^{Z \oplus Y}(\beta) \downarrow = \delta(Z, X)$.

Intuitively, Y helps in recognizing X.
We only consider real numbers (i.e. subsets of ω) in this talk from now on.

We denote the closure of $\emptyset$ under relative recognizability without parameters by $\mathcal{C}$, with parameter α by $\mathcal{C}_\alpha$ and with arbitrary parameters by $\mathcal{C}^\infty$.

What is $\mathcal{C}^\infty$, the recognizable closure of $\emptyset$ with parameter-OTMs?

This is a highly non-absolute object: It could be L if $V = L$.

Better question: What can the recognizable closure be? What is it when the universe is 'as large as possible'? The right 'largeness assumption' turns out to be: There is an ω-sound $\omega_1 + 1$-iterable premouse M with a top measure above a Woodin cardinal and projectum ω.
Such a premouse, if it exists, is unique and we denote it as $M_1^\sharp$.

Theorem: If $M_1^\sharp$ exists, then the recognizable closure (for real numbers) of $\emptyset$ under parameter-OTM-recognizability is the set of reals in M_1, i.e. $\mathcal{C}^\infty = M_1 \cap \mathfrak{P}(\omega)$.

What is $\mathcal{C}^\infty$, the recognizable closure of $\emptyset$ with parameter-OTMs?

This is a highly non-absolute object: It could be L if $V = L$.

Better question: What can the recognizable closure be? What is it when the universe is 'as large as possible'? The right 'largeness assumption' turns out to be: There is an ω-sound $\omega_1 + 1$-iterable premouse M with a top measure above a Woodin cardinal and projectum ω.
Such a premouse, if it exists, is unique and we denote it as $M_1^\sharp$.

Theorem: If $M_1^\sharp$ exists, then the recognizable closure (for real numbers) of $\emptyset$ under parameter-OTM-recognizability is the set of reals in M_1, i.e. $\mathcal{C}^\infty = M_1 \cap \mathfrak{P}(\omega)$.

What is $\mathcal{C}^\infty$, the recognizable closure of $\emptyset$ with parameter-OTMs?

This is a highly non-absolute object: It could be L if $V = L$.

Better question: What can the recognizable closure be? What is it when the universe is 'as large as possible'? The right 'largeness assumption' turns out to be: There is an ω-sound $\omega_1 + 1$-iterable premouse M with a top measure above a Woodin cardinal and projectum ω.
Such a premouse, if it exists, is unique and we denote it as $M_1^\sharp$.

Theorem: If $M_1^\sharp$ exists, then the recognizable closure (for real numbers) of $\emptyset$ under parameter-OTM-recognizability is the set of reals in M_1, i.e. $\mathcal{C}^\infty = M_1 \cap \mathfrak{P}(\omega)$.

What is $\mathcal{C}^\infty$, the recognizable closure of $\emptyset$ with parameter-OTMs?

This is a highly non-absolute object: It could be L if $V = L$.

Better question: What can the recognizable closure be? What is it when the universe is 'as large as possible'? The right 'largeness assumption' turns out to be: There is an ω-sound $\omega_1 + 1$-iterable premouse M with a top measure above a Woodin cardinal and projectum ω.
Such a premouse, if it exists, is unique and we denote it as $M_1^\sharp$.

Theorem: If $M_1^\sharp$ exists, then the recognizable closure (for real numbers) of $\emptyset$ under parameter-OTM-recognizability is the set of reals in M_1, i.e. $\mathcal{C}^\infty = M_1 \cap \mathfrak{P}(\omega)$.

What is $\mathcal{C}^{\infty}$, the recognizable closure of $\emptyset$ with parameter-OTMs?

This is a highly non-absolute object: It could be L if $V = L$.

Better question: What can the recognizable closure be? What is it when the universe is 'as large as possible'? The right 'largeness assumption' turns out to be: There is an ω-sound $\omega_1 + 1$-iterable premouse M with a top measure above a Woodin cardinal and projectum ω.

Such a premouse, if it exists, is unique and we denote it as $M_1^{\sharp}$.

Theorem: If $M_1^{\sharp}$ exists, then the recognizable closure (for real numbers) of $\emptyset$ under parameter-OTM-recognizability is the set of reals in M_1, i.e. $\mathcal{C}^{\infty} = M_1 \cap \mathfrak{P}(\omega)$.

What is $\mathcal{C}^\infty$, the recognizable closure of $\emptyset$ with parameter-OTMs?

This is a highly non-absolute object: It could be L if $V = L$.

Better question: What can the recognizable closure be? What is it when the universe is 'as large as possible'? The right 'largeness assumption' turns out to be: There is an ω-sound $\omega_1 + 1$-iterable premouse M with a top measure above a Woodin cardinal and projectum ω.
Such a premouse, if it exists, is unique and we denote it as $M_1^\sharp$.

Theorem: If $M_1^\sharp$ exists, then the recognizable closure (for real numbers) of $\emptyset$ under parameter-OTM-recognizability is the set of reals in M_1, i.e. $\mathcal{C}^\infty = M_1 \cap \mathfrak{P}(\omega)$.

What is $\mathcal{C}^\infty$, the recognizable closure of $\emptyset$ with parameter-OTMs?

This is a highly non-absolute object: It could be L if $V = L$.

Better question: What can the recognizable closure be? What is it when the universe is 'as large as possible'? The right 'largeness assumption' turns out to be: There is an ω-sound $\omega_1 + 1$-iterable premouse M with a top measure above a Woodin cardinal and projectum ω.
Such a premouse, if it exists, is unique and we denote it as $M_1^\sharp$.

Theorem: If $M_1^\sharp$ exists, then the recognizable closure (for real numbers) of $\emptyset$ under parameter-OTM-recognizability is the set of reals in M_1, i.e. $\mathcal{C}^\infty = M_1 \cap \mathfrak{P}(\omega)$.

What is $\mathcal{C}^\infty$, the recognizable closure of $\emptyset$ with parameter-OTMs?

This is a highly non-absolute object: It could be L if $V = L$.

Better question: What can the recognizable closure be? What is it when the universe is 'as large as possible'? The right 'largeness assumption' turns out to be: There is an ω-sound $\omega_1 + 1$-iterable premouse M with a top measure above a Woodin cardinal and projectum ω.
Such a premouse, if it exists, is unique and we denote it as $M_1^\sharp$.

Theorem: If $M_1^\sharp$ exists, then the recognizable closure (for real numbers) of $\emptyset$ under parameter-OTM-recognizability is the set of reals in M_1, i.e. $\mathcal{C}^\infty = M_1 \cap \mathfrak{P}(\omega)$.

Questions:

- What is the recognizable hull for subsets of arbitrary ordinals?
- What is the recognizable hull relative to the class of cardinals?
- How far can we get even beyond M_1?

Questions:

- What is the recognizable hull for subsets of arbitrary ordinals?
- What is the recognizable hull relative to the class of cardinals?
- How far can we get even beyond M_1?

Questions:

- What is the recognizable hull for subsets of arbitrary ordinals?
- What is the recognizable hull relative to the class of cardinals?
- How far can we get even beyond M_1?

Questions:

- What is the recognizable hull for subsets of arbitrary ordinals?
- What is the recognizable hull relative to the class of cardinals?
- How far can we get even beyond M_1?

Questions:

- What is the recognizable hull for subsets of arbitrary ordinals?
- What is the recognizable hull relative to the class of cardinals?
- How far can we get even beyond M_1?

Variants

Let $x \subseteq \omega$. We say that x is 'semirecognizable' if and only if there is a program P such that, for all $y \subseteq \omega$, $P^y \downarrow$ if and only if $y = x$; that is, P semidecides $\{x\}$.

x is 'corecognizable' or 'antirecognizable' if and only if there is a program P such that, for all $y \subseteq \omega$, $P^y \uparrow$ if and only if $y = x$, i.e. $\mathfrak{P}(\omega) \setminus \{x\}$ is semi-decidable.

x is called 'eventually recognizable' or 'limit recognizable' if and only if there is a program P such that, for all $y \subseteq \omega$, the output of P^y eventually stabilizes at $\delta(x, y)$ (but the computation may not halt).

Variants

Let $x \subseteq \omega$. We say that x is 'semirecognizable' if and only if there is a program P such that, for all $y \subseteq \omega$, $P^y \downarrow$ if and only if $y = x$; that is, P semidecides $\{x\}$.

x is 'corecognizable' or 'antirecognizable' if and only if there is a program P such that, for all $y \subseteq \omega$, $P^y \uparrow$ if and only if $y = x$, i.e. $\mathfrak{P}(\omega) \setminus \{x\}$ is semi-decidable.

x is called 'eventually recognizable' or 'limit recognizable' if and only if there is a program P such that, for all $y \subseteq \omega$, the output of P^y eventually stabilizes at $\delta(x, y)$ (but the computation may not halt).

Variants

Let $x \subseteq \omega$. We say that x is 'semirecognizable' if and only if there is a program P such that, for all $y \subseteq \omega$, $P^y \downarrow$ if and only if $y = x$; that is, P semidecides $\{x\}$.

x is 'corecognizable' or 'antirecognizable' if and only if there is a program P such that, for all $y \subseteq \omega$, $P^y \uparrow$ if and only if $y = x$, i.e. $\mathfrak{P}(\omega) \setminus \{x\}$ is semi-decidable.

x is called 'eventually recognizable' or 'limit recognizable' if and only if there is a program P such that, for all $y \subseteq \omega$, the output of P^y eventually stabilizes at $\delta(x, y)$ (but the computation may not halt).

Let $x \subseteq \omega$, and let $M \in \{\text{ITRM, ITTM, OTM, OTM}\}$.

(i) x is M-recognizable if and only if x M-semirecognizable and M-antirecognizable.

(ii) If x is M-semirecognizable or M-antirecognizable, then x is M-limit recognizable.

(iii) For ITRMs, all of these notions coincide.

Let $x \subseteq \omega$, and let $M \in \{\text{ITRM, ITTM, OTM, OTM}\}$.

(i) x is M-recognizable if and only if x M-semirecognizable and M-antirecognizable.

(ii) If x is M-semirecognizable or M-antirecognizable, then x is M-limit recognizable.

(iii) For ITRMs, all of these notions coincide.

Theorem:[C., Schlicht] There is a real number x that is ITTM-semirecognizable, but not ITTM-recognizable. In fact, the tape content of the universal ITTM $\mathcal{U}$ at time Σ has this property.

Questions: Is there a real number that is ITTM-antirecognizable, but not ITTM-recognizable? Is there a real number that is ITTM-limit recognizable, but neither ITTM-semirecognizable nor ITTM-antirecognizable?

Theorem:[C., Schlicht] There is a real number x that is ITTM-semirecognizable, but not ITTM-recognizable. In fact, the tape content of the universal ITTM $\mathcal{U}$ at time Σ has this property.

Questions: Is there a real number that is ITTM-antirecognizable, but not ITTM-recognizable? Is there a real number that is ITTM-limit recognizable, but neither ITTM-semirecognizable nor ITTM-antirecognizable?

4: RANDOMNESS (joint work with Philipp Schlicht)

A theorem of Sacks

Theorem: Let $f : \mathbb{N} \to \mathbb{N}$. Then there is a program P such that $\{x | \forall i \in \omega P^x(i) \downarrow = f(i)\}$ has positive Lebesgue measure iff f is recursive.

Intuitively: No information from randomness.

A theorem of Sacks

Theorem: Let $f : \mathbb{N} \to \mathbb{N}$. Then there is a program P such that $\{x | \forall i \in \omega P^x(i) \downarrow = f(i)\}$ has positive Lebesgue measure iff f is recursive.

Intuitively: No information from randomness.

In L, the analogue statement for parameter-free OTMs fails in the worst possible way:

Theorem: (C.) Suppose that $V = L$. There is a real x and a co-countable set $A \subseteq {}^{\omega}2$ such that x is OTM-computable without ordinal parameters from every $y \in A$, but x is not OTM-computable without parameters.

However, that is not the whole story...

In L, the analogue statement for parameter-free OTMs fails in the worst possible way:
Theorem: (C.) Suppose that $V = L$. There is a real x and a co-countable set $A \subseteq {}^{\omega}2$ such that x is OTM-computable without ordinal parameters from every $y \in A$, but x is not OTM-computable without parameters.
However, that is not the whole story...

Parameter-free OTMs in the constructible universe

In L, the analogue statement for parameter-free OTMs fails in the worst possible way:

Theorem: (C.) Suppose that $V = L$. There is a real x and a co-countable set $A \subseteq {}^{\omega}2$ such that x is OTM-computable without ordinal parameters from every $y \in A$, but x is not OTM-computable without parameters.

However, that is not the whole story...

The general picture

"Computability in all elements of a non-meager or a positive set implies computability"...

- holds for weak *ITRM*s
- holds for strong *ITRM*s
- holds for *ITTM*s
- holds for α-TMs when α is admissible up to a certain large countable ordinal γ
- holds for α-TMs when $L_\alpha \models ZFC$ (it is hence consistent that it holds cofinally often in ω_1)
- is independent of *ZFC* for parameter-free *OTM*s (and in particular false in L)
- is independent of *ZFC* for parameter-*OTM*s

The general picture

"Computability in all elements of a non-meager or a positive set implies computability"...

- holds for weak *ITRM*s
- holds for strong *ITRM*s
- holds for *ITTM*s
- holds for α-TMs when α is admissible up to a certain large countable ordinal γ
- holds for α-TMs when $L_\alpha \models ZFC$ (it is hence consistent that it holds cofinally often in ω_1)
- is independent of *ZFC* for parameter-free *OTM*s (and in particular false in L)
- is independent of *ZFC* for parameter-*OTM*s

The general picture

"Computability in all elements of a non-meager or a positive set implies computability"...

- holds for weak *ITRM*s
- holds for strong *ITRM*s
- holds for *ITTM*s
- holds for α-TMs when α is admissible up to a certain large countable ordinal γ
- holds for α-TMs when $L_\alpha \models ZFC$ (it is hence consistent that it holds cofinally often in ω_1)
- is independent of *ZFC* for parameter-free *OTM*s (and in particular false in *L*)
- is independent of *ZFC* for parameter-*OTM*s

The general picture

"Computability in all elements of a non-meager or a positive set implies computability"...

- holds for weak *ITRMs*
- holds for strong *ITRMs*
- holds for *ITTMs*
- holds for α-TMs when α is admissible up to a certain large countable ordinal γ
- holds for α-TMs when $L_\alpha \models ZFC$ (it is hence consistent that it holds cofinally often in ω_1)
- is independent of *ZFC* for parameter-free *OTMs* (and in particular false in L)
- is independent of *ZFC* for parameter-*OTMs*

The general picture

"Computability in all elements of a non-meager or a positive set implies computability"...

- holds for weak *ITRMs*
- holds for strong *ITRMs*
- holds for *ITTMs*
- holds for α-TMs when α is admissible up to a certain large countable ordinal γ
- holds for α-TMs when $L_\alpha \models ZFC$ (it is hence consistent that it holds cofinally often in ω_1)
- is independent of *ZFC* for parameter-free *OTMs* (and in particular false in L)
- is independent of *ZFC* for parameter-*OTMs*

The general picture

"Computability in all elements of a non-meager or a positive set implies computability"...

- holds for weak *ITRM*s
- holds for strong *ITRM*s
- holds for *ITTM*s
- holds for α-TMs when α is admissible up to a certain large countable ordinal γ
- holds for α-TMs when $L_\alpha \models ZFC$ (it is hence consistent that it holds cofinally often in ω_1)
- is independent of *ZFC* for parameter-free *OTM*s (and in particular false in L)
- is independent of *ZFC* for parameter-*OTM*s

The general picture

"Computability in all elements of a non-meager or a positive set implies computability"...

- holds for weak *ITRM*s
- holds for strong *ITRM*s
- holds for *ITTM*s
- holds for α-TMs when α is admissible up to a certain large countable ordinal γ
- holds for α-TMs when $L_\alpha \models ZFC$ (it is hence consistent that it holds cofinally often in ω_1)
- is independent of *ZFC* for parameter-free *OTM*s (and in particular false in *L*)
- is independent of *ZFC* for parameter-*OTM*s

The general picture

"Computability in all elements of a non-meager or a positive set implies computability"...

- holds for weak *ITRM*s
- holds for strong *ITRM*s
- holds for *ITTM*s
- holds for α-TMs when α is admissible up to a certain large countable ordinal γ
- holds for α-TMs when $L_\alpha \models ZFC$ (it is hence consistent that it holds cofinally often in ω_1)
- is independent of *ZFC* for parameter-free *OTM*s (and in particular false in *L*)
- is independent of *ZFC* for parameter-*OTM*s

Idea: 'Recognizability relative to all elements of a large (non-meager, positive measure) set of real numbers implies recognizability'

- This holds for ITRMs (C.)
- This holds for ITTMs (C., Schlicht)
- For OTMs with and without parameters, it is independent of ZFC

Randomness and Recognizability

Idea: 'Recognizability relative to all elements of a large (non-meager, positive measure) set of real numbers implies recognizability'

- This holds for ITRMs (C.)
- This holds for ITTMs (C., Schlicht)
- For OTMs with and without parameters, it is independent of ZFC

Idea: ‘Recognizability relative to all elements of a large (non-meager, positive measure) set of real numbers implies recognizability’

- This holds for ITRMs (C.)
- This holds for ITTMs (C., Schlicht)
- For OTMs with and without parameters, it is independent of ZFC

Randomness and Recognizability

Idea: 'Recognizability relative to all elements of a large (non-meager, positive measure) set of real numbers implies recognizability'

- This holds for ITRMs (C.)
- This holds for ITTMs (C., Schlicht)
- For OTMs with and without parameters, it is independent of ZFC

We can also introduce analogues of Martin-Löf-randomness for infinitary machines. For example, we can say that a real number x is ITRM-random if and only if there is no ITRM-decidable set of measure 0 that contains x, or that x is ITRM-generic when there is no ITRM-decidable meager set containing x and similarly for ITTMs (also with semidecidable sets etc.).
This yields nice connections to forcing. For example, ITRM-generic is equivalent to being Cohen-generic over $L_{\omega_\omega^{CK}}$.

Usually no analogue of Kucera-Gacs: Halting problem not solvable relative to a random real.

Analogues of van Lambalgen's theorem etc. [Joint work with Philipp Schlicht; recent paper by Monin and Angles D'Auriac]

We can also introduce analogues of Martin-Löf-randomness for infinitary machines. For example, we can say that a real number x is ITRM-random if and only if there is no ITRM-decidable set of measure 0 that contains x, or that x is ITRM-generic when there is no ITRM-decidable meager set containing x and similarly for ITTMs (also with semidecidable sets etc.).
This yields nice connections to forcing. For example, ITRM-generic is equivalent to being Cohen-generic over $L_{\omega_\omega^{CK}}$.

Usually no analogue of Kucera-Gacs: Halting problem not solvable relative to a random real.

Analogues of van Lambalgen's theorem etc. [Joint work with Philipp Schlicht; recent paper by Monin and Angles D'Auriac]

We can also introduce analogues of Martin-Löf-randomness for infinitary machines. For example, we can say that a real number x is ITRM-random if and only if there is no ITRM-decidable set of measure 0 that contains x, or that x is ITRM-generic when there is no ITRM-decidable meager set containing x and similarly for ITTMs (also with semidecidable sets etc.).
This yields nice connections to forcing. For example, ITRM-generic is equivalent to being Cohen-generic over $L_{\omega_\omega^{CK}}$.

Usually no analogue of Kucera-Gacs: Halting problem not solvable relative to a random real.

Analogues of van Lambalgen's theorem etc. [Joint work with Philipp Schlicht; recent paper by Monin and Angles D'Auriac]

We can also introduce analogues of Martin-Löf-randomness for infinitary machines. For example, we can say that a real number x is ITRM-random if and only if there is no ITRM-decidable set of measure 0 that contains x, or that x is ITRM-generic when there is no ITRM-decidable meager set containing x and similarly for ITTMs (also with semidecidable sets etc.).
This yields nice connections to forcing. For example, ITRM-generic is equivalent to being Cohen-generic over $L_{\omega_\omega^{CK}}$.

Usually no analogue of Kucera-Gacs: Halting problem not solvable relative to a random real.

Analogues of van Lambalgen's theorem etc. [Joint work with Philipp Schlicht; recent paper by Monin and Angles D'Auriac]

We can also introduce analogues of Martin-Löf-randomness for infinitary machines. For example, we can say that a real number x is ITRM-random if and only if there is no ITRM-decidable set of measure 0 that contains x, or that x is ITRM-generic when there is no ITRM-decidable meager set containing x and similarly for ITTMs (also with semidecidable sets etc.).
This yields nice connections to forcing. For example, ITRM-generic is equivalent to being Cohen-generic over $L_{\omega_\omega^{CK}}$.

Usually no analogue of Kucera-Gacs: Halting problem not solvable relative to a random real.

Analogues of van Lambalgen's theorem etc. [Joint work with Philipp Schlicht; recent paper by Monin and Angles D'Auriac]

We can also introduce analogues of Martin-Löf-randomness for infinitary machines. For example, we can say that a real number x is ITRM-random if and only if there is no ITRM-decidable set of measure 0 that contains x, or that x is ITRM-generic when there is no ITRM-decidable meager set containing x and similarly for ITTMs (also with semidecidable sets etc.).
This yields nice connections to forcing. For example, ITRM-generic is equivalent to being Cohen-generic over $L_{\omega_\omega^{CK}}$.

Usually no analogue of Kucera-Gacs: Halting problem not solvable relative to a random real.

Analogues of van Lambalgen's theorem etc. [Joint work with Philipp Schlicht; recent paper by Monin and Angles D'Auriac]

We can also introduce analogues of Martin-Löf-randomness for infinitary machines. For example, we can say that a real number x is ITRM-random if and only if there is no ITRM-decidable set of measure 0 that contains x, or that x is ITRM-generic when there is no ITRM-decidable meager set containing x and similarly for ITTMs (also with semidecidable sets etc.).
This yields nice connections to forcing. For example, ITRM-generic is equivalent to being Cohen-generic over $L_{\omega_{\omega}^{CK}}$.

Usually no analogue of Kucera-Gacs: Halting problem not solvable relative to a random real.

Analogues of van Lambalgen's theorem etc. [Joint work with Philipp Schlicht; recent paper by Monin and Angles D'Auriac]

We can also introduce analogues of Martin-Löf-randomness for infinitary machines. For example, we can say that a real number x is ITRM-random if and only if there is no ITRM-decidable set of measure 0 that contains x, or that x is ITRM-generic when there is no ITRM-decidable meager set containing x and similarly for ITTMs (also with semidecidable sets etc.).
This yields nice connections to forcing. For example, ITRM-generic is equivalent to being Cohen-generic over $L_{\omega_\omega^{CK}}$.

Usually no analogue of Kucera-Gacs: Halting problem not solvable relative to a random real.

Analogues of van Lambalgen's theorem etc. [Joint work with Philipp Schlicht; recent paper by Monin and Angles D'Auriac]

Many specific open questions. More general:

- Incompressibility has hardly been studied. Analogues of Chaitin's Ω?
- Information content of proper classes?
- Infinite Time Martingales?
- Sacks' theorem for definability?

Many specific open questions. More general:

- Incompressibility has hardly been studied. Analogues of Chaitin's Ω?
- Information content of proper classes?
- Infinite Time Martingales?
- Sacks' theorem for definability?

Many specific open questions. More general:

- Incompressibility has hardly been studied. Analogues of Chaitin's Ω?
- Information content of proper classes?
- Infinite Time Martingales?
- Sacks' theorem for definability?

Many specific open questions. More general:

- Incompressibility has hardly been studied. Analogues of Chaitin's Ω?
- Information content of proper classes?
- Infinite Time Martingales?
- Sacks' theorem for definability?

Many specific open questions. More general:

- Incompressibility has hardly been studied. Analogues of Chaitin's Ω?
- Information content of proper classes?
- Infinite Time Martingales?
- Sacks' theorem for definability?

5: COMPLEXITY

R. Schindler introduced complexity theory for ITTMs. Together with Hamkins and Deolalikar, he showed P$\neq$NP, P$\neq$NP$\cap$co-NP for ITTMs.

Later on: Löwe, Winter, Rin, C.: Complexity theory for OTMs.

R. Schindler introduced complexity theory for ITTMs. Together with Hamkins and Deolalikar, he showed P$\neq$NP, P$\neq$NP$\cap$co-NP for ITTMs.

Later on: Löwe, Winter, Rin, C.: Complexity theory for OTMs.

A few results

For OTMs:

- SAT^∞, satisfiability for infinitary propositional logic, is NP^∞-complete (Rin)
- NP^∞-complete problems are undecidable (C., Rin, Löwe)
- There are NP^∞-problems that are neither P^∞ nor NP^∞-complete. In fact, there are infinitely many ∞-polynomially incomparable such problems. (C.)
- Problems decidable by an OTM with a very strict space bound (from a certain input length on, the cardinality of the scratch tape portion used must be less than the cardinality of the input) are regular in the sense of deterministic ordinal automata (C.)

A few results

For OTMs:

- SAT^{∞}, satisfiability for infinitary propositional logic, is NP^{∞}-complete (Rin)
- NP^{∞}-complete problems are undecidable (C., Rin, Löwe)
- There are NP^{∞}-problems that are neither P^{∞} nor NP^{∞}-complete. In fact, there are infinitely many ∞-polynomially incomparable such problems. (C.)
- Problems decidable by an OTM with a very strict space bound (from a certain input length on, the cardinality of the scratch tape portion used must be less than the cardinality of the input) are regular in the sense of deterministic ordinal automata (C.)

A few results

For OTMs:

- SAT^∞, satisfiability for infinitary propositional logic, is NP^∞-complete (Rin)
- NP^∞-complete problems are undecidable (C., Rin, Löwe)
- There are NP^∞-problems that are neither P^∞ nor NP^∞-complete. In fact, there are infinitely many ∞-polynomially incomparable such problems. (C.)
- Problems decidable by an OTM with a very strict space bound (from a certain input length on, the cardinality of the scratch tape portion used must be less than the cardinality of the input) are regular in the sense of deterministic ordinal automata (C.)

A few results

For OTMs:

- SAT^∞, satisfiability for infinitary propositional logic, is NP^∞-complete (Rin)
- NP^∞-complete problems are undecidable (C., Rin, Löwe)
- There are NP^∞-problems that are neither P^∞ nor NP^∞-complete. In fact, there are infinitely many ∞-polynomially incomparable such problems. (C.)
- Problems decidable by an OTM with a very strict space bound (from a certain input length on, the cardinality of the scratch tape portion used must be less than the cardinality of the input) are regular in the sense of deterministic ordinal automata (C.)

A few results

For OTMs:

- SAT^∞, satisfiability for infinitary propositional logic, is NP^∞-complete (Rin)
- NP^∞-complete problems are undecidable (C., Rin, Löwe)
- There are NP^∞-problems that are neither P^∞ nor NP^∞-complete. In fact, there are infinitely many ∞-polynomially incomparable such problems. (C.)
- Problems decidable by an OTM with a very strict space bound (from a certain input length on, the cardinality of the scratch tape portion used must be less than the cardinality of the input) are regular in the sense of deterministic ordinal automata (C.)

A few results

For OTMs:

- SAT^{∞}, satisfiability for infinitary propositional logic, is NP^{∞}-complete (Rin)
- NP^{∞}-complete problems are undecidable (C., Rin, Löwe)
- There are NP^{∞}-problems that are neither P^{∞} nor NP^{∞}-complete. In fact, there are infinitely many ∞-polynomially incomparable such problems. (C.)
- Problems decidable by an OTM with a very strict space bound (from a certain input length on, the cardinality of the scratch tape portion used must be less than the cardinality of the input) are regular in the sense of deterministic ordinal automata (C.)

Some Questions

The class NP^{∞} is much larger than it first looks. For example, it includes $NEXPTIME^{\infty}$. Try to characterize NP^{∞}.

What are the 'right' complexity classes for problems of 'actual mathematics'? A good yardstick problem may be testing a given partial order for well-foundedness.

Standards for a good input encoding?

Some Questions

The class NP^∞ is much larger than it first looks. For example, it includes $NEXPTIME^\infty$. Try to characterize NP^∞.

What are the 'right' complexity classes for problems of 'actual mathematics'? A good yardstick problem may be testing a given partial order for well-foundedness.

Standards for a good input encoding?

Some Questions

The class NP^{∞} is much larger than it first looks. For example, it includes $\mathrm{NEXPTIME}^{\infty}$. Try to characterize NP^{∞}.

What are the 'right' complexity classes for problems of 'actual mathematics'? A good yardstick problem may be testing a given partial order for well-foundedness.

Standards for a good input encoding?

6: GENERALIZED EFFECTIVE REDUCIBILITY

By a natural coding device, sets of ordinals can be regarded as codes for arbitrary sets:

Definition: Let $A \subseteq \mathrm{On}$ be a set of ordinals. We define $d(A)$, the decoding of A, recursively as follows:
$d(A) := \{d(\{\beta : p(\alpha, \beta) \in A\}) : \exists\gamma p(\alpha, \gamma) \in A\}$.

Definition: Let $F : V \to V$ be a class function. We say that P computes F if, whenever c codes x, then P^c converges with an output that codes $F(y)$.

Definition: Given a Π_2-statement $\phi := \forall x \exists y \psi(x, y)$, we call a (class) function F a canonification of ϕ iff $\forall x \psi(x, F(x))$. ϕ is effective iff it has a computable canonification.

Canonifications can be used as oracles: Whenever a code for a set x has been written on the oracle tape, the oracle command creates a code for $F(x)$ on the same tape.

ϕ_1 is ordinal Weihrauch (oW-) reducible to ϕ_2 if there are programs P and Q such that, whenever F_2 is a canonification of ϕ_2, then $P \circ F \circ Q$ is a canonification of ϕ_1 (where we identify programs with the functions they compute).

Definition: Let $F : V \to V$ be a class function. We say that P computes F if, whenever c codes x, then P^c converges with an output that codes $F(y)$.

Definition: Given a Π_2-statement $\phi := \forall x \exists y \psi(x, y)$, we call a (class) function F a canonification of ϕ iff $\forall x \psi(x, F(x))$. ϕ is effective iff it has a computable canonification.

Canonifications can be used as oracles: Whenever a code for a set x has been written on the oracle tape, the oracle command creates a code for $F(x)$ on the same tape.

ϕ_1 is ordinal Weihrauch (oW-) reducible to ϕ_2 if there are programs P and Q such that, whenever F_2 is a canonification of ϕ_2, then $P \circ F \circ Q$ is a canonification of ϕ_1 (where we identify programs with the functions they compute).

Definition: Let $F : V \to V$ be a class function. We say that P computes F if, whenever c codes x, then P^c converges with an output that codes $F(y)$.

Definition: Given a Π_2-statement $\phi := \forall x \exists y \psi(x, y)$, we call a (class) function F a canonification of ϕ iff $\forall x \psi(x, F(x))$. ϕ is effective iff it has a computable canonification.

Canonifications can be used as oracles: Whenever a code for a set x has been written on the oracle tape, the oracle command creates a code for $F(x)$ on the same tape.

ϕ_1 is ordinal Weihrauch (oW-) reducible to ϕ_2 if there are programs P and Q such that, whenever F_2 is a canonification of ϕ_2, then $P \circ F \circ Q$ is a canonification of ϕ_1 (where we identify programs with the functions they compute).

Definition: Let $F : V \to V$ be a class function. We say that P computes F if, whenever c codes x, then P^c converges with an output that codes $F(y)$.

Definition: Given a Π_2-statement $\phi := \forall x \exists y \psi(x, y)$, we call a (class) function F a canonification of ϕ iff $\forall x \psi(x, F(x))$. ϕ is effective iff it has a computable canonification.

Canonifications can be used as oracles: Whenever a code for a set x has been written on the oracle tape, the oracle command creates a code for $F(x)$ on the same tape.

ϕ_1 is ordinal Weihrauch (oW-) reducible to ϕ_2 if there are programs P and Q such that, whenever F_2 is a canonification of ϕ_2, then $P \circ F \circ Q$ is a canonification of ϕ_1 (where we identify programs with the functions they compute).

Some results (we assume that $0^\sharp$ exists):

Theorem: $0 <_{oW} ZL <_{gW} AC <_{oW} WO$. In fact, WO is $\leq_{oW}$-universal with respect to Π_2-theorems of ZF.

This opens the door to an investigation of the effectivity of set-theoretical Π_2-statements in general, and also their reducibility relations.

A specific open question is this: Hausdorff's maximality principle (HMP) is the claim that every partial order has a maximal chain. We know that HMP$\leq_{oW}$WO, but not whether HMP$<_{oW}$WO nor how HMP relates to AC.

This opens the door to an investigation of the effectivity of set-theoretical Π_2-statements in general, and also their reducibility relations.

A specific open question is this: Hausdorff's maximality principle (HMP) is the claim that every partial order has a maximal chain. We know that HMP$\leq_{oW}$WO, but not whether HMP$<_{oW}$WO nor how HMP relates to AC.

This opens the door to an investigation of the effectivity of set-theoretical Π_2-statements in general, and also their reducibility relations.

A specific open question is this: Hausdorff's maximality principle (HMP) is the claim that every partial order has a maximal chain. We know that HMP$\leq_{oW}$WO, but not whether HMP$<_{oW}$WO nor how HMP relates to AC.

7: PHILOSOPHICAL ASPECTS

[G. Takeuti, Proof Theory] "Let us, to begin with, adopt the standpoint of an "infinite mind", which we suppose can examine infinitely many objects one by one. (...) This approach is in its essence the way in which many working mathematicians conceive of sets. It is fair to say that in modern mathematics many of the arguments concerning sets are carried out along these lines, and the higher (finite) order predicate calculus is a formulation of such an approach to sets."

[G.Takeuti, Proof Theory and Set Theory.] "Foundational problems begin when we realize that we cannot examine infinitely many objects one by one. However, it is very easy for us to imagine an infinite mind which can do so. Actually by working in mathematics we have been building up our intuition on what an infinite mind can do. An infinite mind must be able to operate on infinitely many objects as freely as we operate on finitely many objects."

[G. Takeuti, Proof Theory and Set Theory.] "Modern mathematics is the world of the infinite mind."

[G. Takeuti, Proof Theory] “Let us, to begin with, adopt the standpoint of an ”infinite mind”, which we suppose can examine infinitely many objects one by one. (...) This approach is in its essence the way in which many working mathematicians conceive of sets. It is fair to say that in modern mathematics many of the arguments concerning sets are carried out along these lines, and the higher (finite) order predicate calculus is a formulation of such an approach to sets.”

[G.Takeuti, Proof Theory and Set Theory.] “Foundational problems begin when we realize that we cannot examine infinitely many objects one by one. However, it is very easy for us to imagine an infinite mind which can do so. Actually by working in mathematics we have been building up our intuition on what an infinite mind can do. An infinite mind must be able to operate on infinitely many objects as freely as we operate on finitely many objects.”

[G. Takeuti, Proof Theory and Set Theory.] “Modern mathematics is the world of the infinite mind.”

[G. Takeuti, Proof Theory] "Let us, to begin with, adopt the standpoint of an "infinite mind", which we suppose can examine infinitely many objects one by one. (...) This approach is in its essence the way in which many working mathematicians conceive of sets. It is fair to say that in modern mathematics many of the arguments concerning sets are carried out along these lines, and the higher (finite) order predicate calculus is a formulation of such an approach to sets."
[G.Takeuti, Proof Theory and Set Theory.] "Foundational problems begin when we realize that we cannot examine infinitely many objects one by one. However, it is very easy for us to imagine an infinite mind which can do so. Actually by working in mathematics we have been building up our intuition on what an infinite mind can do. An infinite mind must be able to operate on infinitely many objects as freely as we operate on finitely many objects."
[G. Takeuti, Proof Theory and Set Theory.] "Modern mathematics is the world of the infinite mind."

F. Ramsey, The Foundations of Mathematics, on impredicative definitions:

"The only difference is that, owing to our inability to write propositions of infinite length, which is logically a mere accident, $\forall(\phi).\phi(a)$ cannot cannot, like $p \wedge q$, be elementarily expressed, but must be expressed as the logical product of a set of which it is also a member. If we had infinite ressources and could express all atomic functions (...)"

"In this lies the great advantage of my method over that of `Principia Mathematica`. In `Principia` the range of ϕ is that of functions which can be elementarily expressed, and since $\forall\phi.f(\phi!\hat{z},x)$ cannot be so expressed it cannot be a value of $\phi!$; but I define the values of ϕ not by how they can be expressed by us at all, let alone elementarily, **but only by a being with an infinite symbolic system**."

F. Ramsey, The Foundations of Mathematics, on impredicative definitions:

"The only difference is that, owing to our inability to write propositions of infinite length, which is logically a mere accident, $\forall(\phi).\phi(a)$ cannot cannot, like $p \wedge q$, be elementarily expressed, but must be expressed as the logical product of a set of which it is also a member. If we had infinite ressources and could express all atomic functions (...)"

"In this lies the great advantage of my method over that of `Principia Mathematica`. In `Principia` the range of ϕ is that of functions which can be elementarily expressed, and since $\forall\phi.f(\phi!\hat{z}, x)$ cannot be so expressed it cannot be a value of $\phi!$; but I define the values of ϕ not by how they can be expressed by us at all, let alone elementarily, **but only by a being with an infinite symbolic system**."

F. Ramsey, The Foundations of Mathematics, on impredicative definitions:

"The only difference is that, owing to our inability to write propositions of infinite length, which is logically a mere accident, $\forall(\phi).\phi(a)$ cannot cannot, like $p \wedge q$, be elementarily expressed, but must be expressed as the logical product of a set of which it is also a member. If we had infinite ressources and could express all atomic functions (...)"

"In this lies the great advantage of my method over that of `Principia Mathematica`. In `Principia` the range of ϕ is that of functions which can be elementarily expressed, and since $\forall\phi.f(\phi!\hat{z}, x)$ cannot be so expressed it cannot be a value of $\phi!$; but I define the values of ϕ not by how they can be expressed by us at all, let alone elementarily, **but only by a being with an infinite symbolic system**."

W. Hodges, On the Effectivity of some Field Constructions, on using primitive recursive set functions as a formalization of an intuitive notion of 'generalized effectivity' (my emphases):

'It is not too generous. Every Prim$(x, \mathcal{P}_{<\kappa})$ function **can be computed according to an explicit and finite formula by a being who can search only a bounded distance beyond the contents of the function argument and the parameter** x.'

W. Hodges, On the Effectivity of some Field Constructions, on using primitive recursive set functions as a formalization of an intuitive notion of 'generalized effectivity' (my emphases):

'It is not too generous. Every $\mathrm{Prim}(x, \mathcal{P}_{<\kappa})$ function **can be computed according to an explicit and finite formula by a being who can search only a bounded distance beyond the contents of the function argument and the parameter** x.'

H. Wang, A Logical Journey (attributed to K. Gödel) on the justification of the axioms of ZFC:

"To arrive at the totality of integers involves a jump. Overviewing it presupposes an [idealized] infinite intuition. In the second jump we consider not only the integers as given but also the process of selecting integers as given in intuition. (...) Each selection gives a subset as an object. Taking all possible ways of leaving elements out [of the totality of integers] may be thought of as a *method* for producing these objects. (...) What this idealization - realization of a possibility - means is that we conceive and realize the possibility of a mind which can do it."

H. Wang, A Logical Journey (attributed to K. Gödel) on the justification of the axioms of ZFC:

"To arrive at the totality of integers involves a jump. Overviewing it presupposes an [idealized] infinite intuition. In the second jump we consider not only the integers as given but also the process of selecting integers as given in intuition. (...) Each selection gives a subset as an object. Taking all possible ways of leaving elements out [of the totality of integers] may be thought of as a *method* for producing these objects. (...) What this idealization - realization of a possibility - means is that we conceive and realize the possibility of a mind which can do it."

P. Kitcher, The Nature of Mathematical Knowledge, on a naturalistic approach to set theory based on idealized agency:

"I see no bar to the supposition that the sequence of stages at which sets are formed is highly superdenumerable, that each of the stages corresponds to an instant in the life of the constructive subject, and that the subject's activity is carried out in a medium analogous to time, but far richer than time. (...) The view of the ideal subject as an idealization of ourselves does not lapse when we release the subject from the constraints of our time."

P. Kitcher, The Nature of Mathematical Knowledge, on a naturalistic approach to set theory based on idealized agency:

"I see no bar to the supposition that the sequence of stages at which sets are formed is highly superdenumerable, that each of the stages corresponds to an instant in the life of the constructive subject, and that the subject's activity is carried out in a medium `analogous` to time, but far richer than time. (...) The view of the ideal subject as an idealization of ourselves does not lapse when we release the subject from the constraints of our time."

Idea: Regard OTMs as a formalization of idealized agents, in the same way that Turing modeled finite agents via Turing machines. Thus, argue for and work with an 'Infinitary Church-Turing-Thesis'.

Then analyze the criticism and justification of set-theoretical axioms on the basis of idealized agency with this model.

EXAMPLE 1: Obvious analogue of Kleene's realizability interpretation for set-theoretical statements using OTMs.

Idea: Regard OTMs as a formalization of idealized agents, in the same way that Turing modeled finite agents via Turing machines. Thus, argue for and work with an ‘Infinitary Church-Turing-Thesis’.

Then analyze the criticism and justification of set-theoretical axioms on the basis of idealized agency with this model.

EXAMPLE 1: Obvious analogue of Kleene’s realizability interpretation for set-theoretical statements using OTMs.

Idea: Regard OTMs as a formalization of idealized agents, in the same way that Turing modeled finite agents via Turing machines. Thus, argue for and work with an ‘Infinitary Church-Turing-Thesis’.

Then analyze the criticism and justification of set-theoretical axioms on the basis of idealized agency with this model.

EXAMPLE 1: Obvious analogue of Kleene’s realizability interpretation for set-theoretical statements using OTMs.

Let ϕ, ψ be $\in$-formulas, and let P be an OTM-program, $\alpha \in \mathrm{On}$, $a_0, ..., a_n, b_0, ..., b_m$ sets with codes $c(a_0), ..., c(a_n), c(b_0), ..., c(b_m)$ and R, R' be finite tuples.

1. If ϕ is quantifier-free, then (P, α) realizes $\phi(a_0, ..., a_n)$ if and only if $\phi(a_0, ..., a_n)$ is true (in any transitive sets containing $a_0, ..., a_n$).
2. (R, R') realizes $(\phi(a_0, ..., a_n) \wedge \psi(b_0, ..., b_m))$ if and only if R realizes $\phi(a_0, ..., a_n)$ and R' realizes $\psi(b_0, ..., b_m)$.
3. (i, R) realizes $(\phi(a_0, ..., a_n) \vee \psi(b_0, ..., b_m))$ if and only if $i = 0$ and R realizes $\phi(a_0, ..., a_n)$ or $i = 1$ and R realizes $\psi(b_0, ..., b_m)$.
4. (P, α) realizes $\exists x \phi(x, a_0, ..., a_n)$ if and only if $P(\alpha, c(a_0), ..., c(a_n))$ halts with output $(c(b), R)$ where $c(b)$ codes a set b such that R realizes $\phi(b, a_0, ..., a_n)$.
5. (P, α) realizes $\forall x \phi(x, a_0, ..., a_n)$ if and only if, for every code $c(a)$ for a set a, $P(\alpha, c(a), c(a_0), ..., c(a_n))$ halts with output R such that R realizes $\phi(a, a_0, ..., a_n)$.

In this sense, all axioms of KP are realizable, but the power set axiom, the comprehension axiom, the replacement axiom and the wellordering principle - those usually regarded as 'impredicative' or 'non-constructive' - are not.

Let ϕ, ψ be $\in$-formulas, and let P be an OTM-program, $\alpha \in \mathrm{On}$, $a_0, ..., a_n, b_0, ..., b_m$ sets with codes $c(a_0), ..., c(a_n), c(b_0), ..., c(b_m)$ and R, R' be finite tuples.

1. If ϕ is quantifier-free, then (P, α) realizes $\phi(a_0, ..., a_n)$ if and only if $\phi(a_0, ..., a_n)$ is true (in any transitive sets containing $a_0, ..., a_n$).
2. (R, R') realizes $(\phi(a_0, ..., a_n) \wedge \psi(b_0, ..., b_m))$ if and only if R realizes $\phi(a_0, ..., a_n)$ and R' realizes $\psi(b_0, ..., b_m)$.
3. (i, R) realizes $(\phi(a_0, ..., a_n) \vee \psi(b_0, ..., b_m))$ if and only if $i = 0$ and R realizes $\phi(a_0, ..., a_n)$ or $i = 1$ and R realizes $\psi(b_0, ..., b_m)$.
4. (P, α) realizes $\exists x \phi(x, a_0, ..., a_n)$ if and only if $P(\alpha, c(a_0), ..., c(a_n))$ halts with output $(c(b), R)$ where $c(b)$ codes a set b such that R realizes $\phi(b, a_0, ..., a_n)$.
5. (P, α) realizes $\forall x \phi(x, a_0, ..., a_n)$ if and only if, for every code $c(a)$ for a set a, $P(\alpha, c(a), c(a_0), ..., c(a_n))$ halts with output R such that R realizes $\phi(a, a_0, ..., a_n)$.

In this sense, all axioms of KP are realizable, but the power set axiom, the comprehension axiom, the replacement axiom and the wellordering principle - those usually regarded as 'impredicative' or 'non-constructive' - are not.

Let ϕ, ψ be $\in$-formulas, and let P be an OTM-program, $\alpha \in \text{On}$, $a_0, ..., a_n, b_0, ..., b_m$ sets with codes $c(a_0), ..., c(a_n), c(b_0), ..., c(b_m)$ and R, R' be finite tuples.

1. If ϕ is quantifier-free, then (P, α) realizes $\phi(a_0, ..., a_n)$ if and only if $\phi(a_0, ..., a_n)$ is true (in any transitive sets containing $a_0, ..., a_n$).
2. (R, R') realizes $(\phi(a_0, ..., a_n) \wedge \psi(b_0, ..., b_m))$ if and only if R realizes $\phi(a_0, ..., a_n)$ and R' realizes $\psi(b_0, ..., b_m)$.
3. (i, R) realizes $(\phi(a_0, ..., a_n) \vee \psi(b_0, ..., b_m))$ if and only if $i = 0$ and R realizes $\phi(a_0, ..., a_n)$ or $i = 1$ and R realizes $\psi(b_0, ..., b_m)$.
4. (P, α) realizes $\exists x \phi(x, a_0, ..., a_n)$ if and only if $P(\alpha, c(a_0), ..., c(a_n))$ halts with output $(c(b), R)$ where $c(b)$ codes a set b such that R realizes $\phi(b, a_0, ..., a_n)$.
5. (P, α) realizes $\forall x \phi(x, a_0, ..., a_n)$ if and only if, for every code $c(a)$ for a set a, $P(\alpha, c(a), c(a_0), ..., c(a_n))$ halts with output R such that R realizes $\phi(a, a_0, ..., a_n)$.

In this sense, all axioms of KP are realizable, but the power set axiom, the comprehension axiom, the replacement axiom and the wellordering principle - those usually regarded as 'impredicative' or 'non-constructive' - are not.

Let ϕ, ψ be $\in$-formulas, and let P be an OTM-program, $\alpha \in$ On, $a_0, ..., a_n, b_0, ..., b_m$ sets with codes $c(a_0), ..., c(a_n), c(b_0), ..., c(b_m)$ and R, R' be finite tuples.

1. If ϕ is quantifier-free, then (P, α) realizes $\phi(a_0, ..., a_n)$ if and only if $\phi(a_0, ..., a_n)$ is true (in any transitive sets containing $a_0, ..., a_n$).
2. (R, R') realizes $(\phi(a_0, ..., a_n) \wedge \psi(b_0, ..., b_m))$ if and only if R realizes $\phi(a_0, ..., a_n)$ and R' realizes $\psi(b_0, ..., b_m)$.
3. (i, R) realizes $(\phi(a_0, ..., a_n) \vee \psi(b_0, ..., b_m))$ if and only if $i = 0$ and R realizes $\phi(a_0, ..., a_n)$ or $i = 1$ and R realizes $\psi(b_0, ..., b_m)$.
4. (P, α) realizes $\exists x \phi(x, a_0, ..., a_n)$ if and only if $P(\alpha, c(a_0), ..., c(a_n))$ halts with output $(c(b), R)$ where $c(b)$ codes a set b such that R realizes $\phi(b, a_0, ..., a_n)$.
5. (P, α) realizes $\forall x \phi(x, a_0, ..., a_n)$ if and only if, for every code $c(a)$ for a set a, $P(\alpha, c(a), c(a_0), ..., c(a_n))$ halts with output R such that R realizes $\phi(a, a_0, ..., a_n)$.

In this sense, all axioms of KP are realizable, but the power set axiom, the comprehension axiom, the replacement axiom and the wellordering principle - those usually regarded as 'impredicative' or 'non-constructive' - are not.

Let ϕ, ψ be $\in$-formulas, and let P be an OTM-program, $\alpha \in \mathrm{On}$, $a_0, ..., a_n, b_0, ..., b_m$ sets with codes $c(a_0), ..., c(a_n), c(b_0), ..., c(b_m)$ and R, R' be finite tuples.

1. If ϕ is quantifier-free, then (P, α) realizes $\phi(a_0, ..., a_n)$ if and only if $\phi(a_0, ..., a_n)$ is true (in any transitive sets containing $a_0, ..., a_n$).
2. (R, R') realizes $(\phi(a_0, ..., a_n) \wedge \psi(b_0, ..., b_m))$ if and only if R realizes $\phi(a_0, ..., a_n)$ and R' realizes $\psi(b_0, ..., b_m)$.
3. (i, R) realizes $(\phi(a_0, ..., a_n) \vee \psi(b_0, ..., b_m))$ if and only if $i = 0$ and R realizes $\phi(a_0, ..., a_n)$ or $i = 1$ and R realizes $\psi(b_0, ..., b_m)$.
4. (P, α) realizes $\exists x \phi(x, a_0, ..., a_n)$ if and only if $P(\alpha, c(a_0), ..., c(a_n))$ halts with output $(c(b), R)$ where $c(b)$ codes a set b such that R realizes $\phi(b, a_0, ..., a_n)$.
5. (P, α) realizes $\forall x \phi(x, a_0, ..., a_n)$ if and only if, for every code $c(a)$ for a set a, $P(\alpha, c(a), c(a_0), ..., c(a_n))$ halts with output R such that R realizes $\phi(a, a_0, ..., a_n)$.

In this sense, all axioms of KP are realizable, but the power set axiom, the comprehension axiom, the replacement axiom and the wellordering principle - those usually regarded as 'impredicative' or 'non-constructive' - are not.

Let ϕ, ψ be $\in$-formulas, and let P be an OTM-program, $\alpha \in \text{On}$, $a_0, ..., a_n, b_0, ..., b_m$ sets with codes $c(a_0), ..., c(a_n), c(b_0), ..., c(b_m)$ and R, R' be finite tuples.

1. If ϕ is quantifier-free, then (P, α) realizes $\phi(a_0, ..., a_n)$ if and only if $\phi(a_0, ..., a_n)$ is true (in any transitive sets containing $a_0, ..., a_n$).
2. (R, R') realizes $(\phi(a_0, ..., a_n) \wedge \psi(b_0, ..., b_m))$ if and only if R realizes $\phi(a_0, ..., a_n)$ and R' realizes $\psi(b_0, ..., b_m)$.
3. (i, R) realizes $(\phi(a_0, ..., a_n) \vee \psi(b_0, ..., b_m))$ if and only if $i = 0$ and R realizes $\phi(a_0, ..., a_n)$ or $i = 1$ and R realizes $\psi(b_0, ..., b_m)$.
4. (P, α) realizes $\exists x \phi(x, a_0, ..., a_n)$ if and only if $P(\alpha, c(a_0), ..., c(a_n))$ halts with output $(c(b), R)$ where $c(b)$ codes a set b such that R realizes $\phi(b, a_0, ..., a_n)$.
5. (P, α) realizes $\forall x \phi(x, a_0, ..., a_n)$ if and only if, for every code $c(a)$ for a set a, $P(\alpha, c(a), c(a_0), ..., c(a_n))$ halts with output R such that R realizes $\phi(a, a_0, ..., a_n)$.

In this sense, all axioms of KP are realizable, but the power set axiom, the comprehension axiom, the replacement axiom and the wellordering principle - those usually regarded as 'impredicative' or 'non-constructive' - are not.

Let ϕ, ψ be $\in$-formulas, and let P be an OTM-program, $\alpha \in \text{On}$, $a_0, ..., a_n, b_0, ..., b_m$ sets with codes $c(a_0), ..., c(a_n), c(b_0), ..., c(b_m)$ and R, R' be finite tuples.

1. If ϕ is quantifier-free, then (P, α) realizes $\phi(a_0, ..., a_n)$ if and only if $\phi(a_0, ..., a_n)$ is true (in any transitive sets containing $a_0, ..., a_n$).
2. (R, R') realizes $(\phi(a_0, ..., a_n) \wedge \psi(b_0, ..., b_m))$ if and only if R realizes $\phi(a_0, ..., a_n)$ and R' realizes $\psi(b_0, ..., b_m)$.
3. (i, R) realizes $(\phi(a_0, ..., a_n) \vee \psi(b_0, ..., b_m))$ if and only if $i = 0$ and R realizes $\phi(a_0, ..., a_n)$ or $i = 1$ and R realizes $\psi(b_0, ..., b_m)$.
4. (P, α) realizes $\exists x \phi(x, a_0, ..., a_n)$ if and only if $P(\alpha, c(a_0), ..., c(a_n))$ halts with output $(c(b), R)$ where $c(b)$ codes a set b such that R realizes $\phi(b, a_0, ..., a_n)$.
5. (P, α) realizes $\forall x \phi(x, a_0, ..., a_n)$ if and only if, for every code $c(a)$ for a set a, $P(\alpha, c(a), c(a_0), ..., c(a_n))$ halts with output R such that R realizes $\phi(a, a_0, ..., a_n)$.

In this sense, all axioms of KP are realizable, but the power set axiom, the comprehension axiom, the replacement axiom and the wellordering principle - those usually regarded as 'impredicative' or 'non-constructive' - are not.

EXAMPLE 2: Kripke semantics for intuitionistic logic.

Let $s \in \{0,1\}^{**}$, let $\phi(x_0,...,x_n), \psi(y_0,...,y_m)$ be $\in$-formulas and $a_0,...,a_n, b_0,...,b_m$ sets. We define the relation $s \Vdash \phi$ ('s forces ϕ') by recursion as follows:

- If ϕ is Δ_0, then $s \Vdash \phi(a_0,...,a_n)$ if and only if $a_0,...,a_n \in L[s]$ and $L[s] \models \phi(a_0,...,a_n)$.
- $s \Vdash \neg\phi$ if and only if $t \nVdash \phi$ for all $t \supseteq s$.
- $s \Vdash \phi \wedge \psi$ if and only if $s \Vdash \phi$ and $s \Vdash \psi$.
- $s \Vdash \phi \vee \psi$ if and only if $s \Vdash \phi$ or $s \Vdash \psi$.
- $s \Vdash \phi \rightarrow \psi$ if and only if, for all $t \supseteq s$, if $t \Vdash \phi$, then $t \Vdash \psi$.
- $s \Vdash \exists x\phi$ if and only if, $s \Vdash \phi(a)$ for some $a \in L[s]$.
- $s \Vdash \forall x\phi$ if and only if, for all $t \supseteq s$ and all $a \in L[t]$, we have $t \Vdash \phi(a)$.

EXAMPLE 2: Kripke semantics for intuitionistic logic.

Let $s \in \{0,1\}^{**}$, let $\phi(x_0, ..., x_n), \psi(y_0, ..., y_m)$ be $\in$-formulas and $a_0, ..., a_n, b_0, ..., b_m$ sets. We define the relation $s \Vdash \phi$ ('s forces ϕ') by recursion as follows:

- If ϕ is Δ_0, then $s \Vdash \phi(a_0, ..., a_n)$ if and only if $a_0, ..., a_n \in L[s]$ and $L[s] \models \phi(a_0, ..., a_n)$.
- $s \Vdash \neg\phi$ if and only if $t \nVdash \phi$ for all $t \supseteq s$.
- $s \Vdash \phi \wedge \psi$ if and only if $s \Vdash \phi$ and $s \Vdash \psi$.
- $s \Vdash \phi \vee \psi$ if and only if $s \Vdash \phi$ or $s \Vdash \psi$.
- $s \Vdash \phi \rightarrow \psi$ if and only if, for all $t \supseteq s$, if $t \Vdash \phi$, then $t \Vdash \psi$.
- $s \Vdash \exists x\phi$ if and only if, $s \Vdash \phi(a)$ for some $a \in L[s]$.
- $s \Vdash \forall x\phi$ if and only if, for all $t \supseteq s$ and all $a \in L[t]$, we have $t \Vdash \phi(a)$.

EXAMPLE 2: Kripke semantics for intuitionistic logic.

Let $s \in \{0,1\}^{**}$, let $\phi(x_0, ..., x_n), \psi(y_0, ..., y_m)$ be $\in$-formulas and $a_0, ..., a_n, b_0, ..., b_m$ sets. We define the relation $s \Vdash \phi$ ('s forces ϕ') by recursion as follows:

- If ϕ is Δ_0, then $s \Vdash \phi(a_0, ..., a_n)$ if and only if $a_0, ..., a_n \in L[s]$ and $L[s] \models \phi(a_0, ..., a_n)$.
- $s \Vdash \neg\phi$ if and only if $t \nVdash \phi$ for all $t \supseteq s$.
- $s \Vdash \phi \wedge \psi$ if and only if $s \Vdash \phi$ and $s \Vdash \psi$.
- $s \Vdash \phi \vee \psi$ if and only if $s \Vdash \phi$ or $s \Vdash \psi$.
- $s \Vdash \phi \rightarrow \psi$ if and only if, for all $t \supseteq s$, if $t \Vdash \phi$, then $t \Vdash \psi$.
- $s \Vdash \exists x\phi$ if and only if, $s \Vdash \phi(a)$ for some $a \in L[s]$.
- $s \Vdash \forall x\phi$ if and only if, for all $t \supseteq s$ and all $a \in L[t]$, we have $t \Vdash \phi(a)$.

EXAMPLE 2: Kripke semantics for intuitionistic logic.

Let $s \in \{0,1\}^{**}$, let $\phi(x_0, ..., x_n), \psi(y_0, ..., y_m)$ be $\in$-formulas and $a_0, ..., a_n, b_0, ..., b_m$ sets. We define the relation $s \Vdash \phi$ ('s forces ϕ') by recursion as follows:

- If ϕ is Δ_0, then $s \Vdash \phi(a_0, ..., a_n)$ if and only if $a_0, ..., a_n \in L[s]$ and $L[s] \models \phi(a_0, ..., a_n)$.
- $s \Vdash \neg\phi$ if and only if $t \nVdash \phi$ for all $t \supseteq s$.
- $s \Vdash \phi \wedge \psi$ if and only if $s \Vdash \phi$ and $s \Vdash \psi$.
- $s \Vdash \phi \vee \psi$ if and only if $s \Vdash \phi$ or $s \Vdash \psi$.
- $s \Vdash \phi \rightarrow \psi$ if and only if, for all $t \supseteq s$, if $t \Vdash \phi$, then $t \Vdash \psi$.
- $s \Vdash \exists x\phi$ if and only if, $s \Vdash \phi(a)$ for some $a \in L[s]$.
- $s \Vdash \forall x\phi$ if and only if, for all $t \supseteq s$ and all $a \in L[t]$, we have $t \Vdash \phi(a)$.

EXAMPLE 2: Kripke semantics for intuitionistic logic.

Let $s \in \{0,1\}^{**}$, let $\phi(x_0,...,x_n), \psi(y_0,...,y_m)$ be $\in$-formulas and $a_0,...,a_n, b_0,...,b_m$ sets. We define the relation $s \Vdash \phi$ ('s forces ϕ') by recursion as follows:

- If ϕ is Δ_0, then $s \Vdash \phi(a_0,...,a_n)$ if and only if $a_0,...,a_n \in L[s]$ and $L[s] \models \phi(a_0,...,a_n)$.
- $s \Vdash \neg\phi$ if and only if $t \nVdash \phi$ for all $t \supseteq s$.
- $s \Vdash \phi \wedge \psi$ if and only if $s \Vdash \phi$ and $s \Vdash \psi$.
- $s \Vdash \phi \vee \psi$ if and only if $s \Vdash \phi$ or $s \Vdash \psi$.
- $s \Vdash \phi \rightarrow \psi$ if and only if, for all $t \supseteq s$, if $t \Vdash \phi$, then $t \Vdash \psi$.
- $s \Vdash \exists x\phi$ if and only if, $s \Vdash \phi(a)$ for some $a \in L[s]$.
- $s \Vdash \forall x\phi$ if and only if, for all $t \supseteq s$ and all $a \in L[t]$, we have $t \Vdash \phi(a)$.

EXAMPLE 2: Kripke semantics for intuitionistic logic.

Let $s \in \{0,1\}^{**}$, let $\phi(x_0, ..., x_n), \psi(y_0, ..., y_m)$ be $\in$-formulas and $a_0, ..., a_n, b_0, ..., b_m$ sets. We define the relation $s \Vdash \phi$ ('s forces ϕ') by recursion as follows:

- If ϕ is Δ_0, then $s \Vdash \phi(a_0, ..., a_n)$ if and only if $a_0, ..., a_n \in L[s]$ and $L[s] \models \phi(a_0, ..., a_n)$.
- $s \Vdash \neg\phi$ if and only if $t \nVdash \phi$ for all $t \supseteq s$.
- $s \Vdash \phi \wedge \psi$ if and only if $s \Vdash \phi$ and $s \Vdash \psi$.
- $s \Vdash \phi \vee \psi$ if and only if $s \Vdash \phi$ or $s \Vdash \psi$.
- $s \Vdash \phi \rightarrow \psi$ if and only if, for all $t \supseteq s$, if $t \Vdash \phi$, then $t \Vdash \psi$.
- $s \Vdash \exists x\phi$ if and only if, $s \Vdash \phi(a)$ for some $a \in L[s]$.
- $s \Vdash \forall x\phi$ if and only if, for all $t \supseteq s$ and all $a \in L[t]$, we have $t \Vdash \phi(a)$.

EXAMPLE 2: Kripke semantics for intuitionistic logic.

Let $s \in \{0,1\}^{**}$, let $\phi(x_0, ..., x_n), \psi(y_0, ..., y_m)$ be $\in$-formulas and $a_0, ..., a_n, b_0, ..., b_m$ sets. We define the relation $s \Vdash \phi$ ('s forces ϕ') by recursion as follows:

- If ϕ is Δ_0, then $s \Vdash \phi(a_0, ..., a_n)$ if and only if $a_0, ..., a_n \in L[s]$ and $L[s] \models \phi(a_0, ..., a_n)$.
- $s \Vdash \neg\phi$ if and only if $t \nVdash \phi$ for all $t \supseteq s$.
- $s \Vdash \phi \wedge \psi$ if and only if $s \Vdash \phi$ and $s \Vdash \psi$.
- $s \Vdash \phi \vee \psi$ if and only if $s \Vdash \phi$ or $s \Vdash \psi$.
- $s \Vdash \phi \rightarrow \psi$ if and only if, for all $t \supseteq s$, if $t \Vdash \phi$, then $t \Vdash \psi$.
- $s \Vdash \exists x\phi$ if and only if, $s \Vdash \phi(a)$ for some $a \in L[s]$.
- $s \Vdash \forall x\phi$ if and only if, for all $t \supseteq s$ and all $a \in L[t]$, we have $t \Vdash \phi(a)$.

EXAMPLE 2: Kripke semantics for intuitionistic logic.

Let $s \in \{0,1\}^{**}$, let $\phi(x_0, ..., x_n), \psi(y_0, ..., y_m)$ be $\in$-formulas and $a_0, ..., a_n, b_0, ..., b_m$ sets. We define the relation $s \Vdash \phi$ ('s forces ϕ') by recursion as follows:

- If ϕ is Δ_0, then $s \Vdash \phi(a_0, ..., a_n)$ if and only if $a_0, ..., a_n \in L[s]$ and $L[s] \models \phi(a_0, ..., a_n)$.
- $s \Vdash \neg\phi$ if and only if $t \nVdash \phi$ for all $t \supseteq s$.
- $s \Vdash \phi \wedge \psi$ if and only if $s \Vdash \phi$ and $s \Vdash \psi$.
- $s \Vdash \phi \vee \psi$ if and only if $s \Vdash \phi$ or $s \Vdash \psi$.
- $s \Vdash \phi \rightarrow \psi$ if and only if, for all $t \supseteq s$, if $t \Vdash \phi$, then $t \Vdash \psi$.
- $s \Vdash \exists x\phi$ if and only if, $s \Vdash \phi(a)$ for some $a \in L[s]$.
- $s \Vdash \forall x\phi$ if and only if, for all $t \supseteq s$ and all $a \in L[t]$, we have $t \Vdash \phi(a)$.

EXAMPLE 2: Kripke semantics for intuitionistic logic.

Let $s \in \{0,1\}^{**}$, let $\phi(x_0,...,x_n), \psi(y_0,...,y_m)$ be $\in$-formulas and $a_0,...,a_n,b_0,...,b_m$ sets. We define the relation $s \Vdash \phi$ ('s forces ϕ') by recursion as follows:

- If ϕ is Δ_0, then $s \Vdash \phi(a_0,...,a_n)$ if and only if $a_0,...,a_n \in L[s]$ and $L[s] \models \phi(a_0,...,a_n)$.
- $s \Vdash \neg\phi$ if and only if $t \nVdash \phi$ for all $t \supseteq s$.
- $s \Vdash \phi \wedge \psi$ if and only if $s \Vdash \phi$ and $s \Vdash \psi$.
- $s \Vdash \phi \vee \psi$ if and only if $s \Vdash \phi$ or $s \Vdash \psi$.
- $s \Vdash \phi \rightarrow \psi$ if and only if, for all $t \supseteq s$, if $t \Vdash \phi$, then $t \Vdash \psi$.
- $s \Vdash \exists x\phi$ if and only if, $s \Vdash \phi(a)$ for some $a \in L[s]$.
- $s \Vdash \forall x\phi$ if and only if, for all $t \supseteq s$ and all $a \in L[t]$, we have $t \Vdash \phi(a)$.

In this sense, all axioms of KP are forced by $\emptyset$, while the power set axiom and some instances of comprehension or replacement cannot be forced by any string.

Since all sets are constructed, they are naturally well-ordered by their construction ordering, and $\emptyset \Vdash$WO.
(Picture: Realizability semantics: Sets are 'given', the agent gets to know them effectively;
Kripke semantics: Sets are 'constructed' relative to a 'free choice sequence', $s \Vdash \phi$ means that, on the basis of having constructed s alone, the agent can be sure that ϕ will hold in the end.

QUESTION: Axiomatize (under appropriate largeness assumptions on V) those $\in$-sentences that are forced by $\emptyset$/by any string s. (These would correspond to those sentences that the agent can always be sure off and those that he can 'come to know'). The background logic must be intuitionistic, as e.g. both notions violate excluded middle (but one easily checks that the inference rules of intuitionistic logic are valid).

In this sense, all axioms of KP are forced by $\emptyset$, while the power set axiom and some instances of comprehension or replacement cannot be forced by any string.

Since all sets are constructed, they are naturally well-ordered by their construction ordering, and $\emptyset \Vdash$WO.

(Picture: Realizability semantics: Sets are 'given', the agent gets to know them effectively;

Kripke semantics: Sets are 'constructed' relative to a 'free choice sequence', $s \Vdash \phi$ means that, on the basis of having constructed s alone, the agent can be sure that ϕ will hold in the end.

QUESTION: Axiomatize (under appropriate largeness assumptions on V) those $\in$-sentences that are forced by $\emptyset$/by any string s. (These would correspond to those sentences that the agent can always be sure off and those that he can 'come to know'). The background logic must be intuitionistic, as e.g. both notions violate excluded middle (but one easily checks that the inference rules of intuitionistic logic are valid).

In this sense, all axioms of KP are forced by $\emptyset$, while the power set axiom and some instances of comprehension or replacement cannot be forced by any string.

Since all sets are constructed, they are naturally well-ordered by their construction ordering, and $\emptyset \Vdash$WO.
(Picture: Realizability semantics: Sets are 'given', the agent gets to know them effectively;
Kripke semantics: Sets are 'constructed' relative to a 'free choice sequence', $s \Vdash \phi$ means that, on the basis of having constructed s alone, the agent can be sure that ϕ will hold in the end.

QUESTION: Axiomatize (under appropriate largeness assumptions on V) those $\in$-sentences that are forced by $\emptyset$/by any string s. (These would correspond to those sentences that the agent can always be sure off and those that he can 'come to know'). The background logic must be intuitionistic, as e.g. both notions violate excluded middle (but one easily checks that the inference rules of intuitionistic logic are valid).

In this sense, all axioms of KP are forced by $\emptyset$, while the power set axiom and some instances of comprehension or replacement cannot be forced by any string.

Since all sets are constructed, they are naturally well-ordered by their construction ordering, and $\emptyset \Vdash$WO.
(Picture: Realizability semantics: Sets are 'given', the agent gets to know them effectively;
Kripke semantics: Sets are 'constructed' relative to a 'free choice sequence', $s \Vdash \phi$ means that, on the basis of having constructed s alone, the agent can be sure that ϕ will hold in the end.

QUESTION: Axiomatize (under appropriate largeness assumptions on V) those $\in$-sentences that are forced by $\emptyset$/by any string s. (These would correspond to those sentences that the agent can always be sure off and those that he can 'come to know'). The background logic must be intuitionistic, as e.g. both notions violate excluded middle (but one easily checks that the inference rules of intuitionistic logic are valid).

In this sense, all axioms of KP are forced by $\emptyset$, while the power set axiom and some instances of comprehension or replacement cannot be forced by any string.

Since all sets are constructed, they are naturally well-ordered by their construction ordering, and $\emptyset \Vdash$WO.
(Picture: Realizability semantics: Sets are 'given', the agent gets to know them effectively;
Kripke semantics: Sets are 'constructed' relative to a 'free choice sequence', $s \Vdash \phi$ means that, on the basis of having constructed s alone, the agent can be sure that ϕ will hold in the end.

QUESTION: Axiomatize (under appropriate largeness assumptions on V) those $\in$-sentences that are forced by $\emptyset$/by any string s.
(These would correspond to those sentences that the agent can always be sure off and those that he can 'come to know'). The background logic must be intuitionistic, as e.g. both notions violate excluded middle (but one easily checks that the inference rules of intuitionistic logic are valid).

In this sense, all axioms of KP are forced by $\emptyset$, while the power set axiom and some instances of comprehension or replacement cannot be forced by any string.

Since all sets are constructed, they are naturally well-ordered by their construction ordering, and $\emptyset \Vdash$WO.
(Picture: Realizability semantics: Sets are 'given', the agent gets to know them effectively;
Kripke semantics: Sets are 'constructed' relative to a 'free choice sequence', $s \Vdash \phi$ means that, on the basis of having constructed s alone, the agent can be sure that ϕ will hold in the end.

QUESTION: Axiomatize (under appropriate largeness assumptions on V) those $\in$-sentences that are forced by $\emptyset$/by any string s. (These would correspond to those sentences that the agent can always be sure off and those that he can 'come to know'). The background logic must be intuitionistic, as e.g. both notions violate excluded middle (but one easily checks that the inference rules of intuitionistic logic are valid).

In this sense, all axioms of KP are forced by $\emptyset$, while the power set axiom and some instances of comprehension or replacement cannot be forced by any string.

Since all sets are constructed, they are naturally well-ordered by their construction ordering, and $\emptyset \Vdash$WO.
(Picture: Realizability semantics: Sets are 'given', the agent gets to know them effectively;
Kripke semantics: Sets are 'constructed' relative to a 'free choice sequence', $s \Vdash \phi$ means that, on the basis of having constructed s alone, the agent can be sure that ϕ will hold in the end.

QUESTION: Axiomatize (under appropriate largeness assumptions on V) those $\in$-sentences that are forced by $\emptyset$/by any string s. (These would correspond to those sentences that the agent can always be sure off and those that he can 'come to know'). The background logic must be intuitionistic, as e.g. both notions violate excluded middle (but one easily checks that the inference rules of intuitionistic logic are valid).

In this sense, all axioms of KP are forced by $\emptyset$, while the power set axiom and some instances of comprehension or replacement cannot be forced by any string.

Since all sets are constructed, they are naturally well-ordered by their construction ordering, and $\emptyset \Vdash$WO.
(Picture: Realizability semantics: Sets are 'given', the agent gets to know them effectively;
Kripke semantics: Sets are 'constructed' relative to a 'free choice sequence', $s \Vdash \phi$ means that, on the basis of having constructed s alone, the agent can be sure that ϕ will hold in the end.

QUESTION: Axiomatize (under appropriate largeness assumptions on V) those $\in$-sentences that are forced by $\emptyset$/by any string s. (These would correspond to those sentences that the agent can always be sure off and those that he can 'come to know'). The background logic must be intuitionistic, as e.g. both notions violate excluded middle (but one easily checks that the inference rules of intuitionistic logic are valid).

In this sense, all axioms of KP are forced by $\emptyset$, while the power set axiom and some instances of comprehension or replacement cannot be forced by any string.

Since all sets are constructed, they are naturally well-ordered by their construction ordering, and $\emptyset \Vdash$WO.
(Picture: Realizability semantics: Sets are 'given', the agent gets to know them effectively;
Kripke semantics: Sets are 'constructed' relative to a 'free choice sequence', $s \Vdash \phi$ means that, on the basis of having constructed s alone, the agent can be sure that ϕ will hold in the end.

QUESTION: Axiomatize (under appropriate largeness assumptions on V) those $\in$-sentences that are forced by $\emptyset$/by any string s. (These would correspond to those sentences that the agent can always be sure off and those that he can 'come to know'). The background logic must be intuitionistic, as e.g. both notions violate excluded middle (but one easily checks that the inference rules of intuitionistic logic are valid).

In this sense, all axioms of KP are forced by $\emptyset$, while the power set axiom and some instances of comprehension or replacement cannot be forced by any string.

Since all sets are constructed, they are naturally well-ordered by their construction ordering, and $\emptyset \Vdash$WO.
(Picture: Realizability semantics: Sets are 'given', the agent gets to know them effectively;
Kripke semantics: Sets are 'constructed' relative to a 'free choice sequence', $s \Vdash \phi$ means that, on the basis of having constructed s alone, the agent can be sure that ϕ will hold in the end.

QUESTION: Axiomatize (under appropriate largeness assumptions on V) those $\in$-sentences that are forced by $\emptyset$/by any string s.
(These would correspond to those sentences that the agent can always be sure off and those that he can 'come to know'). The background logic must be intuitionistic, as e.g. both notions violate excluded middle (but one easily checks that the inference rules of intuitionistic logic are valid).

Due to time contraints, we skipped...

- Using ITTM-reducibility as a substitute for Borel reducibility (Hamkins, Coskey)
- Infinite time computable model theory (Hamkins, Miller e.a.)
- Degree theory for infinite time machines (Hamkins, Miller, Welch, Dawson, Koepke, Seyfferth, C., ...)
- ...

Due to time contraints, we skipped...

- Using ITTM-reducibility as a substitute for Borel reducibility (Hamkins, Coskey)
- Infinite time computable model theory (Hamkins, Miller e.a.)
- Degree theory for infinite time machines (Hamkins, Miller, Welch, Dawson, Koepke, Seyfferth, C., ...)
- ...

Due to time contraints, we skipped...

- Using ITTM-reducibility as a substitute for Borel reducibility (Hamkins, Coskey)
- Infinite time computable model theory (Hamkins, Miller e.a.)
- Degree theory for infinite time machines (Hamkins, Miller, Welch, Dawson, Koepke, Seyfferth, C., ...)
- ...

Due to time contraints, we skipped...

- Using ITTM-reducibility as a substitute for Borel reducibility (Hamkins, Coskey)
- Infinite time computable model theory (Hamkins, Miller e.a.)
- Degree theory for infinite time machines (Hamkins, Miller, Welch, Dawson, Koepke, Seyfferth, C., ...)
- ...

Due to time contraints, we skipped...

- Using ITTM-reducibility as a substitute for Borel reducibility (Hamkins, Coskey)
- Infinite time computable model theory (Hamkins, Miller e.a.)
- Degree theory for infinite time machines (Hamkins, Miller, Welch, Dawson, Koepke, Seyfferth, C., ...)
- ...

Thank you!

...and of course, I am trying to sell you something...

Thank you!

...and of course, I am trying to sell you something...