Ordinal Computability

Peter Koepke, University of Bonn

CiE 2009, Heidelberg, Germany, July 23, 2009

Ordinal Computability

$1 - 1 + 1 - 1 = 0$

Peter Koepke, University of Bonn

CiE 2009, Heidelberg, Germany, July 23, 2009

Ordinal Computability

$1 - 1 + 1 - 1 = 0$

$\longleftarrow \quad \infty \quad \longrightarrow \quad ?$

Peter Koepke, University of Bonn

CiE 2009, Heidelberg, Germany, July 23, 2009

Ordinal Computability

$$1 - 1 + 1 - 1 = 0$$

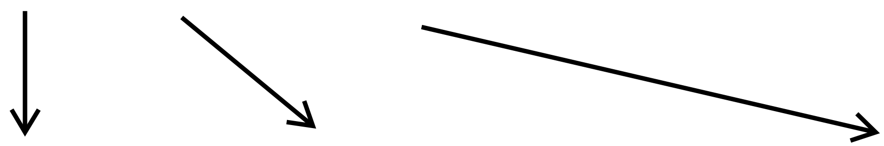
$$1 - 1 + 1 - 1 + \ldots\ldots = \text{limit}\ldots$$

Peter Koepke, University of Bonn

CiE 2009, Heidelberg, Germany, July 23, 2009

Ordinal Computability

$$1 - 1 + 1 - 1 = 0$$



$$1 - 1 + 1 - 1 + \ldots\ldots = 0$$

Peter Koepke, University of Bonn

CiE 2009, Heidelberg, Germany, July 23, 2009

Ordinal Computability

$$1 - 1 + 1 - 1 = 0$$

$$1 - 1 + 1 - 1 + \ldots\ldots + 1 + 2 + \ldots = ?$$

Peter Koepke, University of Bonn

CiE 2009, Heidelberg, Germany, July 23, 2009

Ordinal Computability

$$1 - 1 + 1 - 1 = 0$$

$$1 - 1 + \infty - 1 + \ldots\ldots + \infty + 2 + \ldots = \infty^2$$

Peter Koepke, University of Bonn

CiE 2009, Heidelberg, Germany, July 23, 2009

Contents

Effective computability

- f is recursive
- f is finitely definable
- f is Herbrand-Gödel computable
- f is representable in a consistent formal system $\supseteq \mathcal{R}$
- f is Turing computable
- f is flowchart (or “while”) computable
- f is λ-definable

Rôles of natural numbers

- finite number of steps in calculations and deductions
- finite contents of memory in computations
- finite size of programs, recursion schemas, λ-terms, etc.
- algebraic properties: 0, $n+1$
- order properties: $m<n$
- induction and recursion

Other algebraic domains or orders

- continuous time $\mathbb{R}$
- numbers and data from other rings and fields
- ordinal numbers
- .....

Other algebraic domains or orders

- continuous time $\mathbb{R}$
- numbers and data from other rings and fields
- <u>ordinal numbers</u>
- .....

Ordinals

- ”counting unboundedly”
- finite ordinals = natural numbers: $0, 1, 2, 3, \ldots, n, \ldots$
- endextend by limits: $0, 1, 2, 3, \ldots, n, \ldots, \infty$
- endextend by successors: $0, 1, 2, 3, \ldots, n, \ldots, \infty, \infty + 1$

Ordinals

$0, 1, 2, 3, ..., n, ...$

Ordinals

$0, 1, 2, 3, ..., n, ...$

ω

Ordinals

$0, 1, 2, 3, \ldots, n, \ldots$

$\omega, \omega + 1$

Ordinals

$0, 1, 2, 3, \ldots, n, \ldots$

$\omega, \omega + 1, \omega + 2$

Ordinals

$0, 1, 2, 3, ..., n, ...$

$\omega, \omega + 1, \omega + 2, ...$

Ordinals

$0, 1, 2, 3, \ldots, n, \ldots$

$\omega, \omega + 1, \omega + 2, \ldots$

$\omega + \omega = \omega \cdot 2$

Ordinals

$$0, 1, 2, 3, \ldots, n, \ldots$$

$$\omega, \omega + 1, \omega + 2, \ldots$$

$$\omega + \omega = \omega \cdot 2, \omega \cdot 2 + 1$$

Ordinals

$0, 1, 2, 3, \ldots, n, \ldots$

$\omega, \omega + 1, \omega + 2, \ldots$

$\omega + \omega = \omega \cdot 2, \omega \cdot 2 + 1, \ldots$

Ordinals

$0, 1, 2, 3, \ldots, n, \ldots$

$\omega, \omega + 1, \omega + 2, \ldots$

$\omega + \omega = \omega \cdot 2, \omega \cdot 2 + 1, \ldots$

$\vdots$

Ordinals

$0, 1, 2, 3, \ldots, n, \ldots$

$\omega, \omega + 1, \omega + 2, \ldots$

$\omega + \omega = \omega \cdot 2, \omega \cdot 2 + 1, \ldots$

$\vdots$

$\omega \cdot \omega, \ldots$

Ordinals

$0, 1, 2, 3, \ldots, n, \ldots$

$\omega, \omega + 1, \omega + 2, \ldots$

$\omega + \omega = \omega \cdot 2, \omega \cdot 2 + 1, \ldots$

$\vdots$

$\omega \cdot \omega, \ldots$

$\vdots$

Ordinals

$0, 1, 2, 3, \ldots, n, \ldots$

$\omega, \omega + 1, \omega + 2, \ldots$

$\omega + \omega = \omega \cdot 2, \omega \cdot 2 + 1, \ldots$

$\vdots$

$\omega \cdot \omega, \ldots$

$\vdots$

$\aleph_1 = \omega_1, \ldots$

Ordinals

$$0, 1, 2, 3, \ldots, n, \ldots$$

$$\omega, \omega + 1, \omega + 2, \ldots$$

$$\omega + \omega = \omega \cdot 2, \omega \cdot 2 + 1, \ldots$$

$$\vdots$$

$$\omega \cdot \omega, \ldots$$

$$\vdots$$

$$\aleph_1 = \omega_1, \ldots$$

$$\vdots$$

Ordinals

- The ordinals form a *proper class* Ord of objects
- the ordinals are linearly ordered
- the ordinals are closed under the $+1$ -operation
- there are *limit* ordinals like $\omega, \omega+\omega, \ldots, \aleph_1, \ldots$
- the ordinals are wellordered, i.e. there is *no* infinite descending chain $\alpha_0 > \alpha_1 > \alpha_2 > \ldots$ of ordinals
- the ordinals are the ordertypes of wellordered sets

Ordinals

- one can do induction and recursion along the ordinals
- ordinal addition is defined by recursion
  - initial case: $\alpha + 0 = \alpha$
  - successor case: $\alpha + (\beta + 1) = (\alpha + \beta) + 1$
  - *limit* case: if β is a limit ordinal then

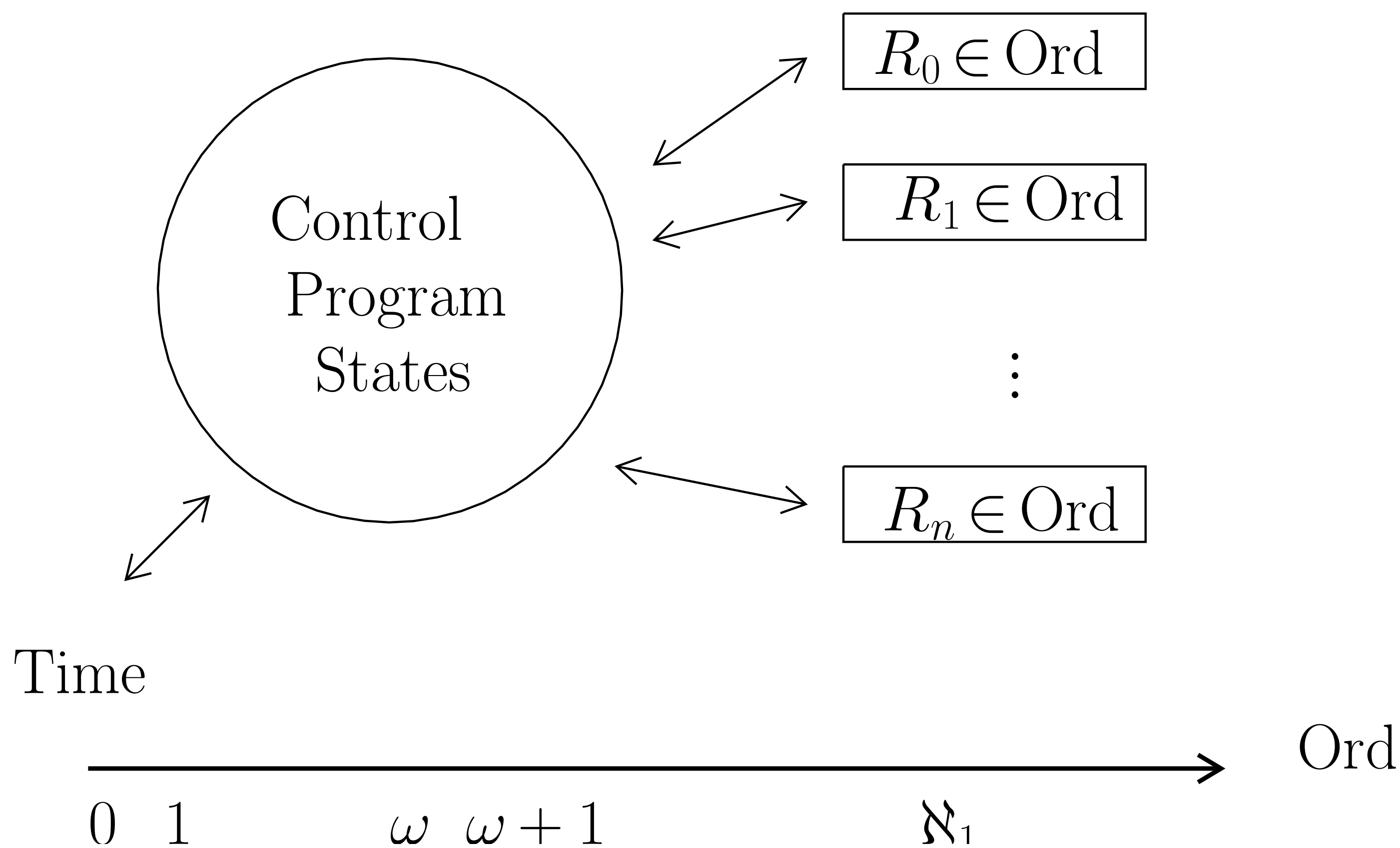
$$\alpha + \beta = \lim_{i < \beta} (\alpha + i)$$

Ordinal register machines

$\alpha + \beta$ can be "computed" as follows:

- put α and β in registers R_0 and R_1
- set a register R_2 to 0
- count up registers R_0 and R_2 in parallel
- stop when R_2 reaches R_1, and output R_0
- at limit "times" let the contents of R_0 and R_2 be the limits of the previous contents

Ordinal register machines



Ordinal register machines, successor times

A *register program* is a finite list $P = I_0, I_1, \ldots, I_{s-1}$ of *instructions*:

- the *zero instruction* $Z(m)$ set register R_m to 0
- the *successor instruction* $S(m)$ increases register R_m by 1
- the *transfer instruction* $T(m, m')$ sets $R_{m'}$ to the contents of R_m
- the *jump instruction* $J(m, m', q)$: if $R_m = R_{m'}$, the register machine proceeds to the qth instruction of P; otherwise it proceeds to the next instruction in P
- the machine halts if the "next instruction" is not in P

Ordinal register machines, limit times

- Let $t \in \mathrm{Ord}$ be a limit "time"
- $\liminf_{s \to t} R_m(s)$ is the smallest ordinal ρ such that $\{s < t \mid R_m(s) \leqslant \rho\}$ is unbounded in t
- at limit times, the machine registers follow the liminf rule

$$R_m(t) = \liminf_{s \to t} R_m(s)$$

- at limit times the program jumps to a specific limit state

ORM computable functions

- $\alpha + \beta$
- $\alpha \cdot \beta$, where $\alpha \cdot 0 = 0$, $\alpha \cdot (\beta + 1) = (\alpha \cdot \beta) + \alpha$, $\alpha \cdot \gamma = \lim_{\beta < \gamma} (\alpha \cdot \beta)$, for limit ordinals γ
- α^β, where $\alpha^0 = 1$, $\alpha^{\beta+1} = \alpha^\beta \cdot \alpha$, $\alpha^\gamma = \lim_{\beta < \gamma} \alpha^\beta$, for limit ordinals γ

ORM computability

- what is the class of ORM computable functions?

- what is the class of ORM computable sets, i.e. the class of sets of the form

$$\{\alpha < \beta | F(\alpha, \vec{\gamma}) = 1\}$$

where F is ORM computable and $\beta, \vec{\gamma} \in \mathrm{Ord}$?

A recursion theorem

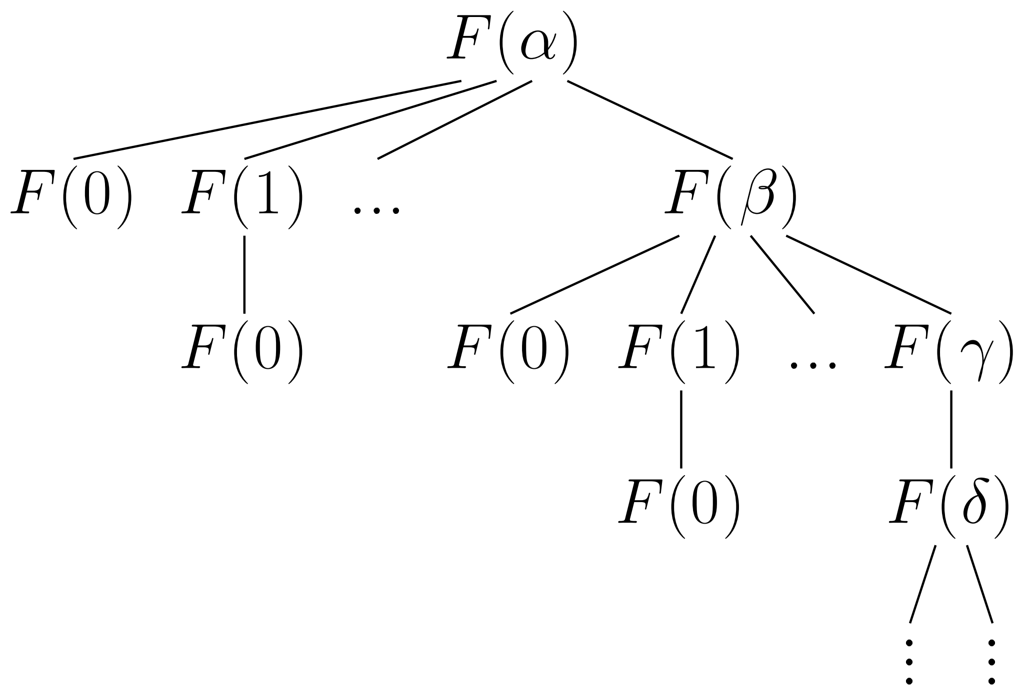
Let $H: \mathrm{Ord}^3 \rightarrow \mathrm{Ord}$ be ORM computable. Define

$$F(\alpha) = \begin{cases} 1 \text{ iff } \exists \nu < \alpha \; H(\alpha, \nu, F(\nu)) = 1 \\ 0 \text{ else} \end{cases}$$

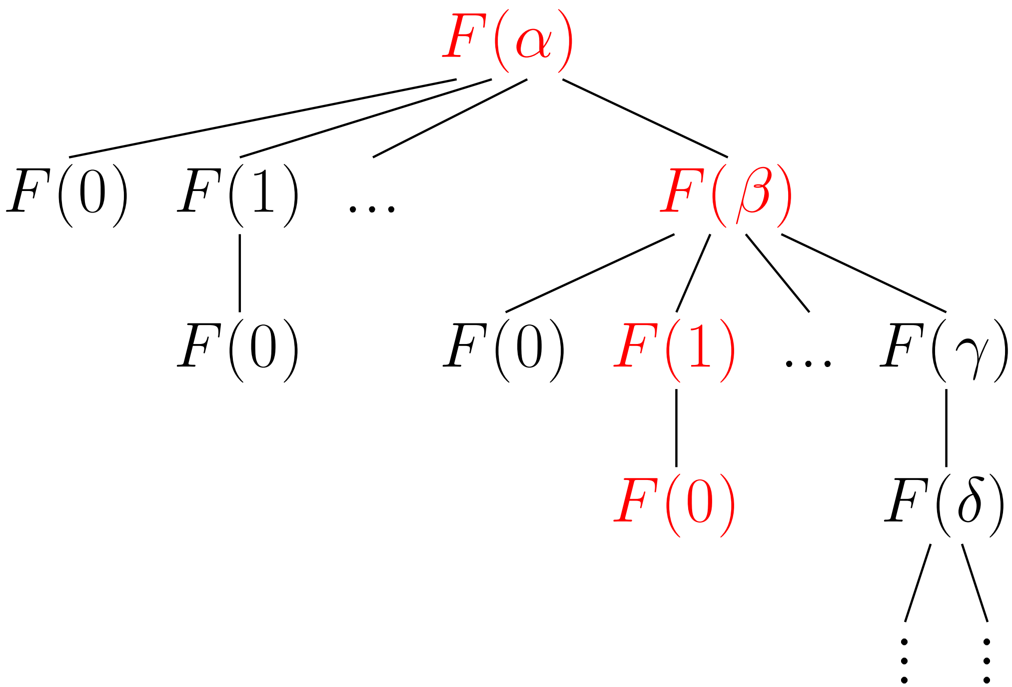
Then $F: \mathrm{Ord} \rightarrow \mathrm{Ord}$ is ORM computable.

A recursion theorem

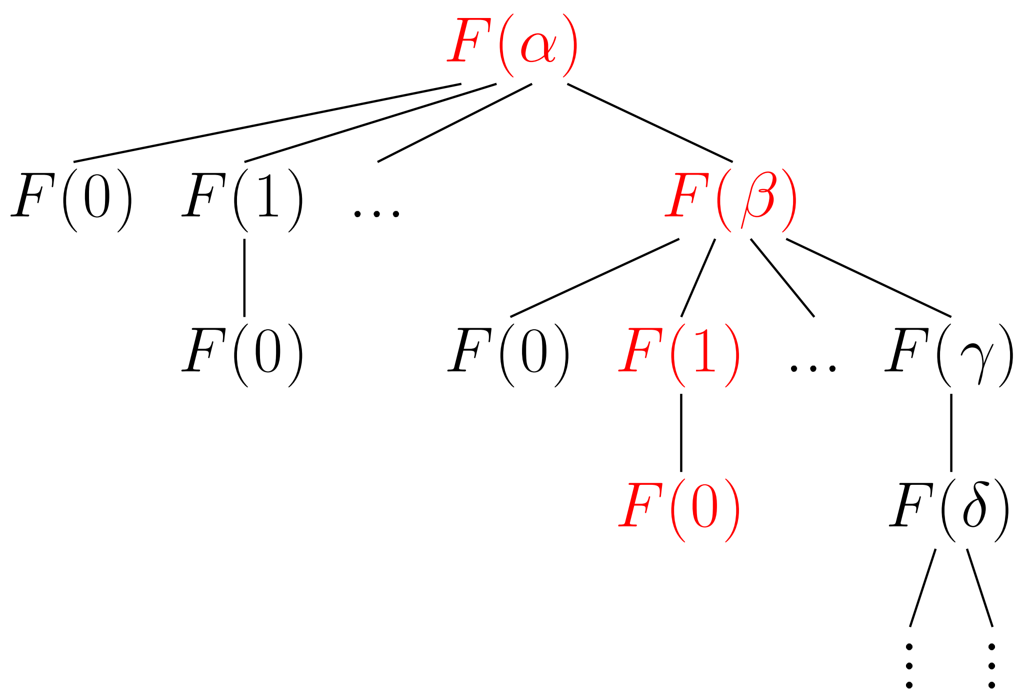
$$F(\alpha) = 1 \text{ iff } \exists \beta < \alpha \; H(\alpha, \beta, F(\beta)) = 1$$



A recursion theorem



A recursion theorem



Search for a good path using a stack $F(\alpha)?, F(\beta)?, F(\gamma)?, \ldots$

A recursion theorem

Code the stack $\alpha_0 > \alpha_1 > \ldots > \alpha_{n-1}$ into one register

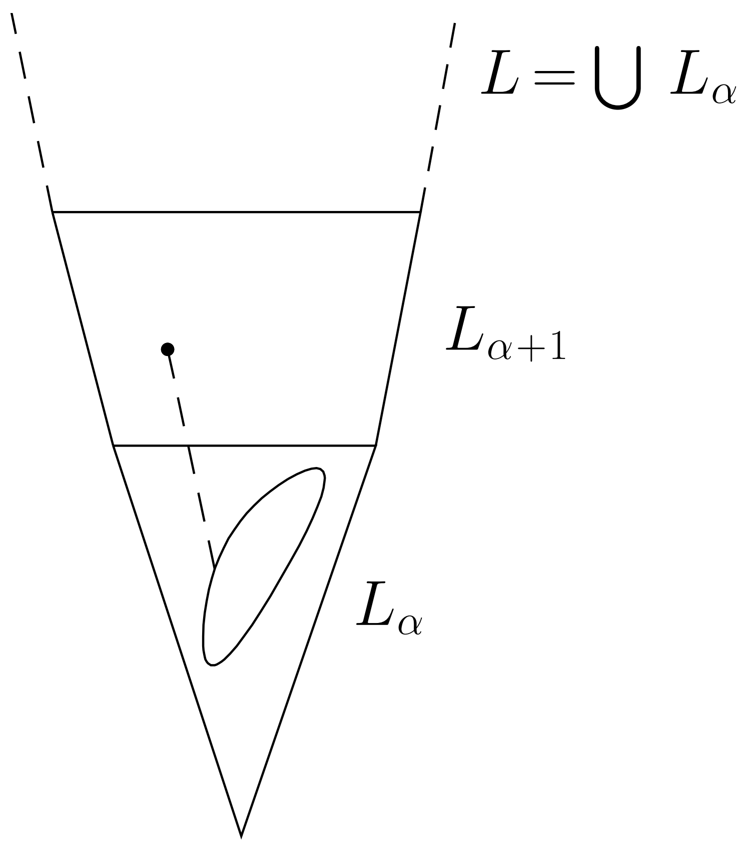
$$R_m = 3^{\alpha_0} + 3^{\alpha_1} + \ldots + 3^{\alpha_{n-2}} + 3^{\alpha_{n-1}}.$$

The constructible model L

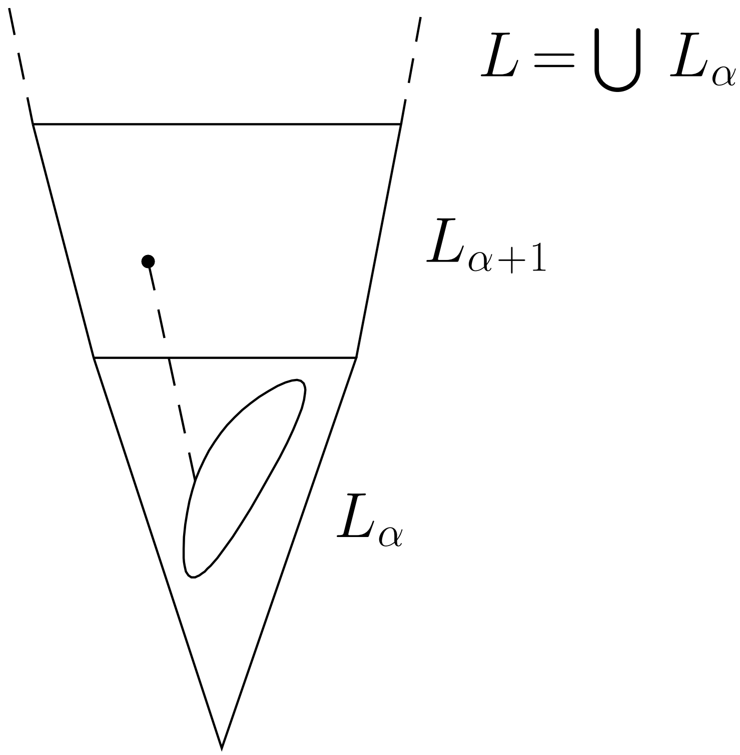
Kurt Gödel defined the following model of the axioms of set theory

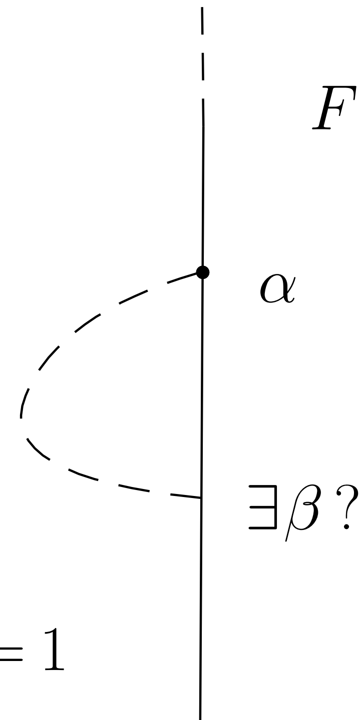
- $L_0 = \emptyset$
- $L_{\alpha+1}$ = the collection of subsets of L_α which are first order definable in the structure $(L_\alpha, \in)$ with parameters
- $L_\gamma = \bigcup_{\alpha<\gamma} L_\alpha$ for limit ordinals γ
- $L = \bigcup_{\alpha \in \mathrm{Ord}} L_\alpha$

The constructible model L



The constructible model L





$F(\alpha) = 1$

iff $\exists \beta < \alpha \; H(\alpha, \beta, F(\beta)) = 1$

The constructible model L

Theorem (_, Siders) A set X of ordinals is ORM computable iff $X \in L$, i.e. if X is *constructible*.

Proof. ($\rightarrow$) Any ORM computation can be carried out inside the model L, hence $X \in L$.

($\leftarrow$) One can code iterated definability and the L_α-hierarchy into the ordinals so that the associated operations become ORM computable. So constructible sets of ordinals are ORM computable.

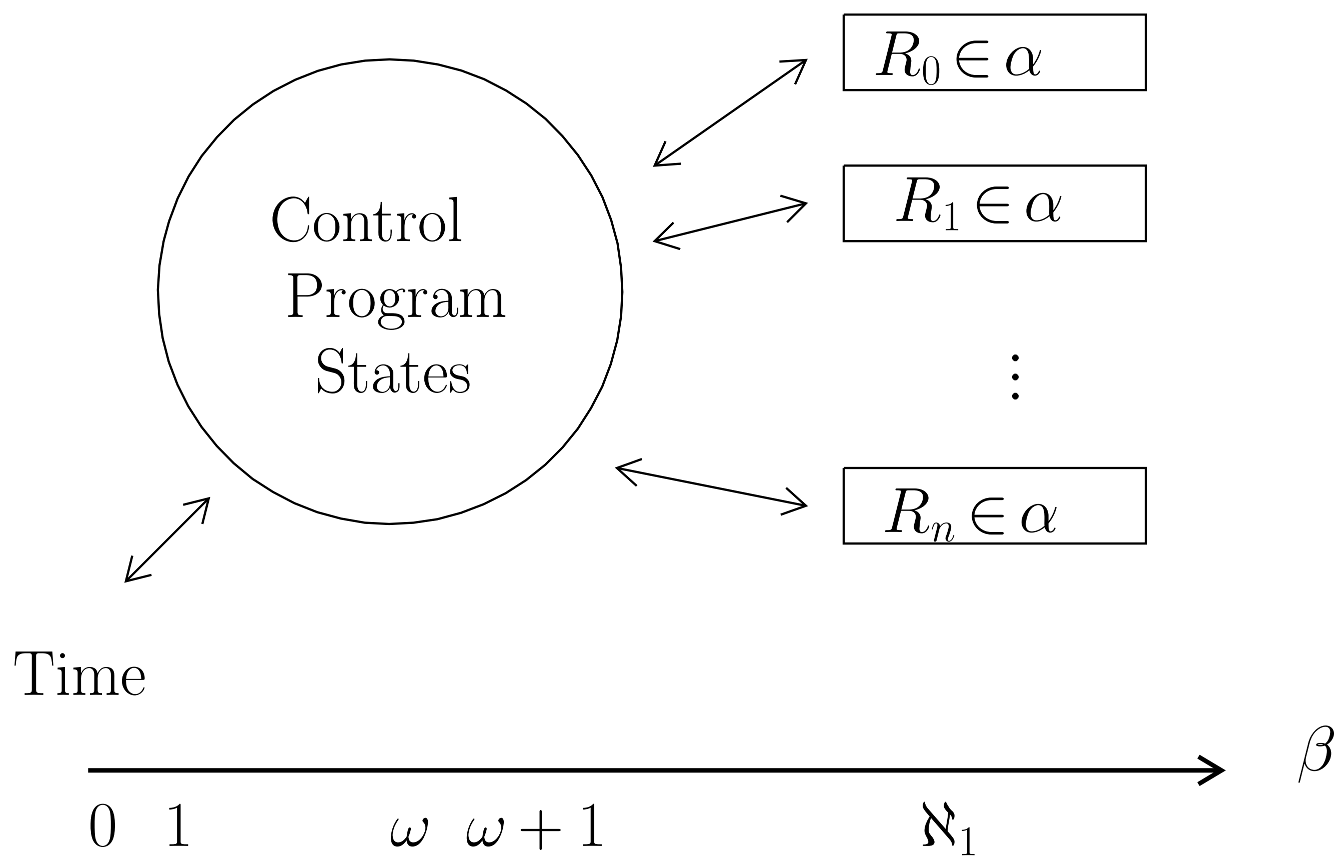
The constructible model L

- Gödel's Axiom of Constructibility can be reformulated as: every set of ordinals is ORM computable

- One can use the computability perspective to prove the Generalised Continuum Hypothesis and other principles in L

- From a universal ORM one can define a "Silver machine" which allows to prove Jensen's finestructural principles in L

α-β-Register machines, space α, time β



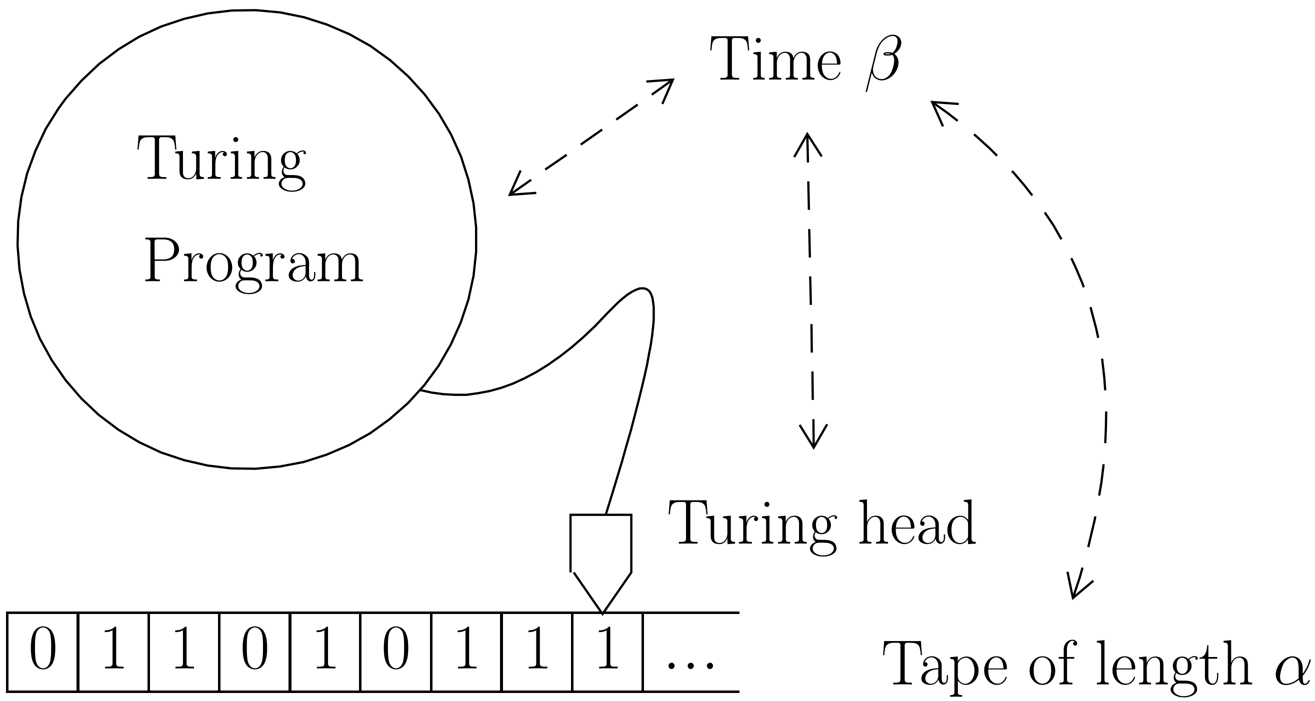
α-β-Register machines

- ω-ω-machines are classical register machines
- ω-Ord-machines are register versions of the Infinite Time Turing Machines (using adequate limit operations)
- for α admissible, α-α-computability corresponds to α-recursion theory (Sacks et al)

α-β-Register machines

Register machines	space ω	space admissible α	space Ord
time ω	standard register machine computable $= \Delta^0_1$	-	-
time admissible α	?	α register machine (α recursion theory) computable $= \mathbf{\Delta}_1(L_\alpha)$ [_, Seyfferth]	-
time Ord	ITRM Infinite time register machine computable $= L_{\omega_\omega^{\mathrm{CK}}} \cap \mathcal{P}(\omega)$ [_, CiE 2009]	?	Ordinal register machine computable $= L \cap \mathcal{P}(\mathrm{Ord})$ [_, Siders]

α-β-Turing machines



α-β-Turing machines

TURING	space ω	space admissible α	space Ord
time ω	standard TURING machine computable $= \Delta_1^0$	-	-
time admissible α	?	α TURING machine (α-recursion theory) computable $= \mathbf{\Delta}_1(L_\alpha)$ [_, Seyfferth]	-
time Ord	ITTM $\mathbf{\Delta}_1^1 \subsetneq$ computable in real parameter $\subsetneq \mathbf{\Delta}_2^1$ [Hamkins et al]	?	Ordinal TURING machine computable $= L \cap \mathcal{P}(\mathrm{Ord})$ [_]

α-β-X machines

For a

- (classical) machine model X
- ordinal space α
- ordinal time β

determine the class of computable sets.

α-β-X machines

For a

- (classical) machine model X
- ordinal space α
- ordinal time β

determine the class of computable sets.

This gives a parametrised spectrum from classical computability theory to constructibility theory, i.e. set theory.

Infinite Time Register Machines, ITRM = ω-Ord-register machines

- use "hardware" of classical register machines
- use arbitrary ordinal time
- use liminf rule with the proviso that at time t register R_m is "reset" to 0 if $\liminf_{s<t} R_m(s) = \omega$

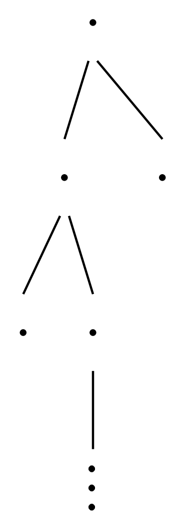
Infinite Time Register Machines

Theorem (_) A real number $a \in {}^{\omega}2$ is computable by an ITRM iff $a \in L_{\omega_\omega^{\mathrm{CK}}}$.

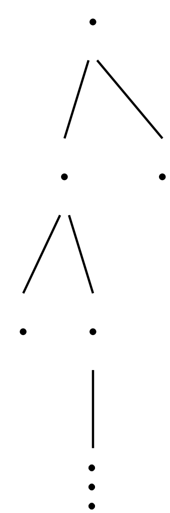
Here ω_0^{CK}, ω_1^{CK}, ..., $\omega_\omega^{\mathrm{CK}}$ is the monotone enumeration of the first admissible ordinals and their limit.

Infinite Time Register Machines

Theorem (_, Miller) The set $\mathrm{WO} = \{Z \in {}^{\omega}2 | Z$ codes a wellorder$\}$ is computable by an ITRM.



Infinite Time Register Machines



Look for an infinite branch in Z, keeping the finite attempts in a register R_m ; if there is an infinite branch, the register will overrun and be reset to 0; otherwise it will not overrun and have a finite liminf.

Infinite Time Register Machines

The *hyperjump* $Z^+ \in {}^\omega 2$ of $Z \in {}^\omega 2$ is defined by:

$Z^+(n) = 1$ iff $\{(i, j) \in \omega \times \omega | P_n^Z(2^i \cdot 3^j) = 1\}$ is a wellfounded relation.

where $P_0, P_1, \ldots$ is a fixed recursive enumeration of all register programs and let $P_n^Z: \omega \rightharpoonup \omega$ be the partial function given by P_n with oracle Z. Then

Theorem. $0, 0^+, 0^{++}, \ldots, 0^{(l)}, \ldots$ are all ITRM computable.

Infinite Time Register Machines

The

Theorem. $0, 0^{+}, 0^{++}, \ldots, 0^{(l)}, \ldots$ are all ITRM computable.

implies:

Theorem. Every real in $L_{\omega_{\omega}^{\mathrm{CK}}}$ is ITRM computable.

Proof. Because every real in $L_{\omega_{\omega}^{\mathrm{CK}}}$ is Turing computable from some finite iterate $0^{(l)}$ of the hyperjump.

Infinite Time Register Machines

Theorem (_) If an ITRM with n registers stops, it will do so before time $\omega_{n+1}^{\mathrm{CK}}$.

Idea. If an ITRM computation runs for $\aleph_1$ many steps then by a downward Löwenheim-Skolem argument there is a closed unbounded sets of ordinals $<\aleph_1$ where the machine configuration is the same as at $\aleph_1$. But then the machine will "cycle" after $\aleph_1$.

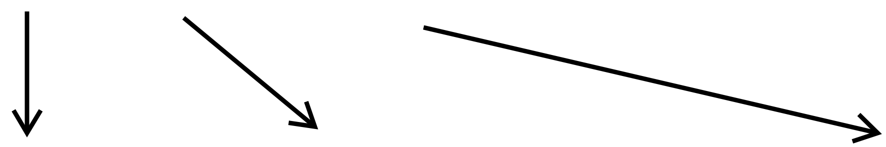
This argument can be refined to work at $\omega_{n+1}^{\mathrm{CK}}$ instead of $\aleph_1$.

So every ITRM computable real a can be computed within $L_{\omega_\omega^{\mathrm{CK}}}$; $a \in L_{\omega_\omega^{\mathrm{CK}}}$.

Ordinal Computability

- analyses various classes of sets by atomic Turing or register operations together with limit operations
- connects classical computability theory, higher recursion theory, descriptive set theory, and constructibility theory
- still has many accessible open problems: certain combinations of space α and time β, other machine models
- has participated at CiE since the first conference at Amsterdam

$$1 - 1 + 1 - 1 = 0$$



$$1 - 1 + \omega - 1 + \ldots\ldots + \omega + 2 + \ldots = \aleph_1^2$$

Thank you!