Ordinals, Computations, and Models of Set Theory

A Tutorial at Days in Logic, Coimbra, Portugal

(preliminary version)

BY PETER KOEPKE

University of Bonn

January 2006

Abstract

Ordinary computations can be characterised by register machines working with natural numbers. We study ordinal register machines where the registers can hold arbitrary ordinal numbers. The class of sets of ordinals which are computable by such machines has strong closure properties and satisfies the set theoretic axiom system SO. This implies that ordinal computability is equivalent to GÖDEL's model L of constructible sets. In this tutorial we shall give a proof of this theorem, starting with brief reviews of ordinal theory and standard register machines.

Register machines

```
Addition, computing gamma = alpha + beta:
0  alpha':=0
1  beta':=0
2  gamma:=0
3  if alpha=alpha' then go to 7
4  alpha':=alpha'+1
5  gamma:=gamma+1
6  go to 3
7  if beta=beta' then STOP
8  beta':=beta'+1
9  gamma:=gamma+1
10 go to 7
```

Working on ordinals instead of natural numbers?!

Theorem 1. *A set* $x \subseteq \mathrm{Ord}$ *is ordinal computable if and only if* $x \in L$.

Questions:

- how can other notions of computability be lifted from natural numbers to ordinals?
- how do recursion theoretic notions lift to ordinal machines, and what is their set-theoretic significance?
- can ordinal machines be used for the fine-structural analysis of the constructible universe?
- can we generate larger models of set theory by some stronger notions of ordinal computation?

Structure of the tutorial:

- A review of the theory of ordinals.
- A short review of standard register machines.
- Definition of ordinal register machines.
- The theory SO of sets of ordinals.
- Interpreting ZFC within SO.
- A recursion theorem for ordinal computability.
- Computing a model of SO.
- Every constructible set of ordinals is ordinal computable.
- An application: the generalised continuum hypothesis in L.

Ordinal numbers

$$\begin{aligned}
0 &= \emptyset \\
1 &= \{0\} \\
2 &= \{0,1\} \\
3 &= \{0,1,2\} \\
&\vdots \\
n+1 &= \{0,1,\dots,n\} \\
&\vdots
\end{aligned}$$

$$m < n \leftrightarrow m \in n$$

Definition 2. *A set or class A is* transitive, $\mathrm{Trans}(A)$, *iff* $\forall u, v\,(u \in v \wedge v \in A \rightarrow u \in A)$.

Definition 3. *A set x is an* ordinal number, $\mathrm{Ord}(x)$, *if* $\mathrm{Trans}(x) \wedge \forall y \in x\, \mathrm{Trans}(y)$. *Let*

$$\mathrm{Ord} = \{x \mid x \textit{ is an ordinal number}\}$$

be the class of all ordinals.

Theorem 4. *a) The class* Ord *is transitive.*

b) Ord *is linearly ordered by* $<$.

c) Ord *is well-ordered by* $<$, *i.e.,*

$$\forall x \subseteq \mathrm{Ord}\,(x \neq \emptyset \rightarrow \exists \alpha \in x \forall \beta < \alpha \beta \notin x).$$

Definition 5. *An ordinal* α *is a* successor ordinal *if it is of the form* $\alpha = \beta + 1$. *An ordinal* α *is a* limit ordinal *if* α *is not a successor ordinal and* $\alpha \neq 0$.

Definition 6. *Let* ω *be the smallest limit ordinal.* ω *is the set of* natural numbers.

Theorem 7. *The structure* $(\omega, 0, +1)$ *satisfies the* PEANO *axioms. In particular the principle of* complete induction *holds:*

$$\forall X \subseteq \omega (0 \in X \wedge \forall n (n \in X \rightarrow n + 1 \in X) \rightarrow X = \omega).$$

Induction and recursion

Theorem 8. *Let $\varphi(v, \vec{w})$ be an $\in$-formula. Then*

$$\forall \vec{w}\, (\exists \alpha \in \mathrm{Ord}\, \varphi(\alpha, \vec{w}) \rightarrow \exists \alpha \in \mathrm{Ord}\, (\varphi(\alpha, \vec{w}) \wedge \forall \beta < \alpha \neg \varphi(\beta, \vec{w}))).$$

Theorem 9. *Let $\varphi(v, \vec{w})$ be an $\in$-formula and assume that*

- $\varphi(0, \vec{w})$
- $\forall \alpha \in \mathrm{Ord}\, (\varphi(\alpha, \vec{w}) \rightarrow \varphi(\alpha + 1, \vec{w}))$
- $\forall \alpha (\alpha \textit{ is a limit ordinal } \rightarrow (\forall \beta < \alpha \varphi(\beta, \vec{w}) \rightarrow \varphi(\alpha, \vec{w})))$

Then $\forall \alpha \in \mathrm{Ord}\, \varphi(\alpha, \vec{w})$.

Theorem 10. *Let $G\colon V \rightarrow V$ be a definable function. Then there is a unique definable function F: $\mathrm{Ord} \rightarrow V$ which for every $\alpha \in \mathrm{Ord}$ satisfies the recursion equation*

$$F(\alpha) = G(F \upharpoonright \alpha).$$

Theorem 11. *Let $G_0 \in V$ and let $G_{\mathrm{succ}}\colon V \rightarrow V$ and $G_{\mathrm{lim}}\colon V \rightarrow V$ be definable functions. Then there is a unique definable function $F\colon \mathrm{Ord} \rightarrow V$ such that*

- $F(0) = G_0$
- $\forall \alpha \in \mathrm{Ord}\ F(\alpha + 1) = G_{\mathrm{succ}}(F(\alpha))$
- $\forall \alpha \in \mathrm{Ord}\,(\alpha \textit{ is a limit ordinal } \rightarrow F(\alpha) = G_{\mathrm{lim}}(F \upharpoonright \alpha))$.

Definition 12. *The* VON NEUMANN *hierarchy* $(V_\alpha \mid \alpha \in \mathrm{Ord})$*:*

- $V_0 = \emptyset$
- $V_{\alpha+1} = \mathcal{P}(V_\alpha)$
- $V_\lambda = \bigcup_{\alpha<\lambda} V_\alpha = \bigcup \mathrm{range}((V_\alpha \mid \alpha < \lambda))$.

Definition 13. *The* ordinal sum $\alpha+\beta$ *is defined by recursion on* $\beta\in\mathrm{Ord}$ *by*

- $\alpha+0=\alpha$
- $\alpha+(\beta+1)=(\alpha+\beta)+1$
- $\alpha+\lambda=\bigcup_{\beta<\lambda}(\alpha+\beta)$.

Definition 14. *The* ordinal product $\alpha\cdot\beta$ *is defined by recursion on* $\beta\in\mathrm{Ord}$ *by*

- $\alpha\cdot 0=0$
- $\alpha\cdot(\beta+1)=(\alpha\cdot\beta)+\alpha$
- $\alpha\cdot\lambda=\bigcup_{\beta<\lambda}(\alpha\cdot\beta)$.

Definition 15. *Let* $(\delta_i\,|\,i<\lambda)$ *be a sequence of ordinals of limit length* λ*. Then*

a) $\lim_{i<\lambda}\delta_i=\bigcup_{i<\lambda}\delta_i$ *is the* limit *of* $(\delta_i\,|\,i<\lambda)$*;*

b) $\liminf_{i<\lambda}\delta_i=\lim_{i<\lambda}\min\{\delta_j\,|\,i\leqslant j<\lambda\}$ *is the* inferior limit *of* $(\delta_i\,|\,i<\lambda)$*.*

The Gödel pairing function

Definition 16. *Define a well-ordering* $\prec$ *on* $\mathrm{Ord}\times\mathrm{Ord}$ *by*

$$\begin{aligned}(\gamma,\delta)\prec(\gamma',\delta') \text{ iff } & \max(\gamma,\delta)<\max(\gamma',\delta')\vee\\ & (\max(\gamma,\delta)=\max(\gamma',\delta')\wedge\gamma<\gamma')\vee\\ & (\max(\gamma,\delta)=\max(\gamma',\delta')\wedge\gamma=\gamma'\wedge\delta<\delta').\end{aligned}$$

Exercise 1. Show that $\prec$ is a set-like well-ordering of $\mathrm{Ord}\times\mathrm{Ord}$. *Set-like* means that

$$\forall\gamma',\delta'\,\{(\gamma,\delta)\,|\,(\gamma,\delta)\prec(\gamma',\delta')\} \text{ is a set.}$$

Definition 17. *Define an order-isomorphism* $G^{-1}\colon\mathrm{Ord}\to\mathrm{Ord}\times\mathrm{Ord}$ *recursively by*

$$G^{-1}(\alpha)=\textit{ the }\prec\textit{-minimal element of }\mathrm{Ord}\times\mathrm{Ord}\setminus\{G^{-1}(\beta)\,|\,\beta<\alpha\}.$$

G is called the Gödel pairing function*.*

Register machines

Definition 18. *An* unlimited register machine *URM has registers* $R_0, R_1, \ldots$ *which can hold* natural numbers. *A register program consists of commands to increase or to reset a register. The program may jump on condition of equality between two registers.*

An URM program *is a finite list* $P = I_0, I_1, \ldots, I_{s-1}$ *of* instructions *each of which may be of one of four kinds:*

a) *the* zero instruction $Z(n)$ *changes the contents of* R_n *to* 0, *leaving all other registers unaltered;*

b) *the* successor instruction $S(n)$ *increases the natural number contained in* R_n *by* 1, *leaving all other registers unaltered;*

c) *the* transfer instruction $T(m,n)$ *replaces the contents of* R_n *by the natural number contained in* R_m, *leaving all other registers unaltered;*

d) *the* jump instruction $J(m,n,q)$ *is carried out within the program* P *as follows: the contents* r_m *and* r_n *of the registers* R_m *and* R_n *are compared, but all the registers are left unaltered; then, if* $R_m = R_n$, *the URM proceeds to the qth instruction of P; if* $R_m \neq R_n$, *the URM proceeds to the next instruction in P.*

The instructions of a register program can be addressed by their indices which are called program states. *At each ordinal time t the machine will be in a configuration consisting of a program state* $I(t) \in \omega$ *and the register contents which can be viewed as a function* $R(t)\colon \omega \to \omega$. $R(t)(n)$ *is the content of the register* R_n *at time* t. *We also write* $R_n(t)$ *instead of* $R(t)(n)$.

Definition 19. *Let $P = I_0, I_1, \ldots, I_{s-1}$ be a program. A triple*

$$I: \theta \rightarrow \omega, R: \theta \rightarrow ({}^{\omega}\omega)$$

is a (register) *computation by P if the following hold:*

a) *$\theta \leqslant \omega$; θ is the* length *of the computation;*

b) *$I(0) = 0$; the machine starts in state 0;*

c) *If $t < \theta$ and $I(t) \notin s = \{0, 1, \ldots, s-1\}$ then $\theta = t + 1$; the machine* stops *if the machine state is not a program state of P;*

d) *If $t < \theta$ and $I(t) \in \mathrm{state}(P)$ then $t + 1 < \theta$; the next configuration is determined by the instruction $I_{I(t)}$:*

   i. *if $I_{I(t)}$ is the zero instruction $Z(n)$ then let $I(t+1) = I(t) + 1$ and define $R(t+1): \omega \rightarrow \mathrm{Ord}$ by*

$$R_k(t+1) = \begin{cases} 0, & \text{if } k = n \\ R_k(t), & \text{if } k \neq n \end{cases}$$

ii. *if* $I_{I(t)}$ *is the successor instruction* $S(n)$ *then let* $I(t+1)=I(t)+1$ *and define* $R(t+1)\colon \omega\rightarrow\mathrm{Ord}$ *by*

$$R_k(t+1)=\begin{cases} R_k(t)+1, & \textit{if } k=n\\ R_k(t), & \textit{if } k\neq n\end{cases}$$

iii. *if* $I_{I(t)}$ *is the transfer instruction* $T(m,n)$ *then let* $I(t+1)=I(t)+1$ *and define* $R(t+1)\colon \omega\rightarrow\mathrm{Ord}$ *by*

$$R_k(t+1)=\begin{cases} R_m(t), & \textit{if } R_m(t)=R_n(t)\\ R_k(t), & \textit{if } R_m(t)\neq R_n(t)\end{cases}$$

iv. *if* $I_{I(t)}$ *is the jump instruction* $J(m,n,q)$ *then let* $R(t+1)=R(t)$ *and*

$$I(t+1)=\begin{cases} q, & \textit{if } k=n\\ I(t)+1, & \textit{if } k\neq n\end{cases}$$

The computation is obviously recursively determined by the initial register contents $R(0)$ *and the program* P*. We call it the* computation by P with impput $R(0)$*. If the computation stops at length* $\theta=\beta+1<\omega$ *then* $R(\beta)$ *are the final register contents. In this case we say that* P computes $R(\beta)(0)$ *from* $R(0)$ *and write* $P\colon R(0)\mapsto R(\beta)(0)$*.*

Algorithms

Addition, computing gamma = alpha + beta:

```
0  alpha':=0
1  beta':=0
2  gamma:=0
3  if alpha=alpha' then go to 7
4  alpha':=alpha'+1
5  gamma:=gamma+1
6  go to 3
7  if beta=beta' then STOP
8  beta':=beta'+1
9  gamma:=gamma+1
10 go to 7
```

Decrementation, computing beta = alpha $\dot{-}$ 1:

```
0  alpha':=0
1  beta:=0
2  if alpha=alpha' then STOP
3  alpha':=alpha'+1
4  if alpha=alpha' then STOP
5  beta:=beta+1
6  go to 3
```

```
Multiplication, computing gamma = alpha * beta:

0  beta':=0

1  gamma:=0

2  if beta=beta' then STOP

3  beta':=beta'+1

4  gamma:=gamma + alpha

5  go to 2
```

Multiplication, computing gamma = alpha * beta:

```
0  beta':=0
1  gamma:=0
2  if beta=beta' then STOP
3  beta'=beta'+1
4    alpha'':=0
5    beta'':=0
6    gamma':=0
7    gamma=alpha'' then go to 11
8    alpha''=alpha''+1
9   gamma'=gamma'+1
10   go to 7
11   if alpha=beta'' then go to 15
12   beta''=beta''+1
13   gamma'=gamma'+1
14   go to 11
15   gamma:=gamma'
16 go to 2
```

Ordinal computations

Definition 20. *Let $P = I_0, I_1, \ldots, I_{s-1}$ be an* URM *program. A triple*

$$I\colon \theta \to \omega,\ R\colon \theta \to ({}^{\omega}\mathrm{Ord})$$

is an (ordinal register) computation *by P if the following hold:*

a) *θ is a successor ordinal or $\theta = \mathrm{Ord}$; θ is the* length *of the computation;*

b) *$I(0) = 0$; the machine starts in state 0;*

c) *If $t < \theta$ and $I(t) \notin s = \{0, 1, \ldots, s-1\}$ then $\theta = t + 1$; the machine* stops *if the machine state is not a program state of P;*

d) *If $t < \theta$ and $I(t) \in \mathrm{state}(P)$ then $t + 1 < \theta$; the next configuration is determined by the instruction $I_{I(t)}$:*

   i. *if $I_{I(t)}$ is the zero instruction $Z(n)$ then let $I(t+1) = I(t) + 1$ and define $R(t+1)\colon \omega \to$* Ord *by*

$$R_k(t+1) = \begin{cases} 0, & \text{if } k = n \\ R_k(t), & \text{if } k \neq n \end{cases}$$

   ii. *if $I_{I(t)}$ is the successor instruction $S(n)$ then let $I(t+1) = I(t) + 1$ and define $R(t+1)$: $\omega \to$* Ord *by*

$$R_k(t+1) = \begin{cases} R_k(t) + 1, & \text{if } k = n \\ R_k(t), & \text{if } k \neq n \end{cases}$$

iii. if $I_{I(t)}$ is the transfer instruction $T(m,n)$ then let $I(t+1)=I(t)+1$ and define $R(t+1)\colon\omega\rightarrow\mathrm{Ord}$ by

$$R_k(t+1)=\begin{cases} R_m(t), & \textit{if } k=n\\ R_k(t), & \textit{if } k\neq n\end{cases}$$

iv. if $I_{I(t)}$ is the jump instruction $J(m,n,q)$ then let $R(t+1)=R(t)$ and

$$I(t+1)=\begin{cases} q, & \textit{if } R_m(t)=R_n(t)\\ I(t)+1, & \textit{if } R_m(t)\neq R_n(t)\end{cases}$$

e) If $t<\theta$ is a limit ordinal, the machine constellation at t is determined by taking inferior limits:

$$\begin{aligned}\forall k\in\omega\, R_k(t) &= \liminf_{r\rightarrow t} R_k(r);\\ I(t) &= \liminf_{r\rightarrow t} I(r).\end{aligned}$$

The computation is obviously determined recursively by the initial register contents $R(0)$ and the program P. We call it the ordinal computation by P with imput $R(0)$. *If the computation stops, $\theta=\beta+1$ is a successor ordinal and $R(\beta)$ is the final register content. In this case we say that P* computes $R(\beta)(0)$ *from $R(0)$ and write $P\colon R(0)\mapsto R(\beta)(0)$.*

```
      ...
⟶ 17:begin loop
        ...
   21:   begin subloop
           ...
   29:   end subloop
        ...
   32:end loop
      ...
```

Assume that for times $r \to t$ the loop $(17 - 32)$ with its subloop $(21 - 29)$ is traversed cofinally often. Then at time t it is natural to put the machine at the start of the "main loop". Assuming that the lines of the program are enumerated in increasing order this corresponds to the liminf rule

$$I(t) = \liminf_{r \to t} S(r).$$

Definition 21. *An n-ary partial function $F\colon \mathrm{Ord}^n \rightharpoonup \mathrm{Ord}$ is* (ordinal register) computable *if there is a register program P such that for every n-tuple $(\alpha_0, \ldots, \alpha_{n-1})$ holds*

$$P\colon (\alpha_0, \ldots, \alpha_{n-1}, 0, 0, \ldots) \mapsto F(\alpha_0, \ldots, \alpha_{n-1}).$$

Definition 22. *A subset $x \subseteq \mathrm{Ord}$ is* (ordinal register) computable *if there is a register program P and ordinals $\delta_1, \ldots, \delta_{n-1}$ such that for every $\alpha \in \mathrm{Ord}$ holds*

$$P\colon (\alpha, \delta_1, \ldots, \delta_{n-1}, 0, 0, \ldots) \mapsto \chi_x(\alpha),$$

where χ_x is the characteristic function of x.

Ordinal algorithms

```
Ordinal addition, computing gamma = alpha + beta:

0  alpha':=0

1  beta':=0

2  gamma:=0

3  if alpha=alpha' then go to 7

4  alpha':=alpha'+1

5  gamma:=gamma+1

6  go to 3

7  if beta=beta' then STOP

8  beta':=beta'+1

9  gamma:=gamma+1

10 go to 7
```

Observe that at limit times this algorithm, by the liminf rule, will nicely cycle back to the beginnings of loops `3 - 6` or `7 - 10` resp. and thus it will implement the recursion rule for addition at limit ordinals.

`Ordinal decrement, computing beta = alpha` $\dot{-}$ `1:`

```
0  alpha':=0

1  beta:=0

2  if alpha=alpha' then STOP

3  alpha':=alpha'+1

4  if alpha=alpha' then STOP

5  beta:=beta+1

6  go to 3
```

Theorem 23. *Let $f(v_0, \ldots, v_{n-1})$ and $g_0(\vec{w}), \ldots, g_{n-1}(\vec{w})$ be computable functions on the ordinals. Then the composition $h(\vec{w}) = f(g_0(\vec{w}), \ldots, g_{n-1}(\vec{w}))$ is ordinal register computable.*

```
Ordinal exponentiation, computing gamma = beta ** alpha:

0  gamma:=1

1  alpha':=0

2  if alpha=alpha' then STOP

3  gamma=gamma * beta

4  alpha'=alpha'+1

5  go to 2
```

Goedel pairing, computing gamma = G(alpha,beta):

```
0  alpha':=0
1  beta':=0
2  eta:=0
3  flag:=0
4  gamma:=0
5  if alpha=alpha' and beta=beta' then STOP
6  if alpha'=eta and and beta'=eta and flag=0 then
      alpha'=0, flag:=1, go to 5 fi
7  if alpha'=eta and and beta'=eta and flag=1 then
      eta:=eta+1, alpha'=eta, beta'=0, gamma:=gamma+1, go to 5 fi
8  if beta'<eta and flag=0 then
      beta':=beta'+1, gamma:=gamma+1, go to 5 fi
9  if alpha'<eta and flag=1 then
      alpha':=alpha'+1, gamma:=gamma+1, go to 5 fi
```

Theorem 24. *Let the function* $f\colon \mathrm{Ord}^n \rightarrow \mathrm{Ord}$ *be ordinal register computable and surjective. Then there are ordinal register computable functions* $g_0, \ldots, g_{n-1}\colon \mathrm{Ord} \rightarrow \mathrm{Ord}$ *such that*

$$\forall \alpha\, f(g_0(\alpha), \ldots, g_{n-1}(\alpha)) = \alpha .$$

The inverse functions G_0 and G_1 satisfying

$$\forall \gamma\, \gamma = G(G_0(\gamma), G_1(\gamma))$$

are ordinal computable as well.

The theory SO

Definition 25. *The Theory* SO *is formulated in the first-order language* L_{SO} *and consists of the following list of axioms:*

1. Well-ordering axiom (WO)*:*
   $\forall \alpha, \beta, \gamma (\neg \alpha < \alpha \wedge (\alpha < \beta \wedge \beta < \gamma \rightarrow \alpha < \gamma) \wedge$
   $(\alpha < \beta \vee \alpha = \beta \vee \beta < \alpha)) \wedge$
   $\forall a (\exists \alpha \alpha \in a \rightarrow \exists \alpha (\alpha \in a \wedge \forall \beta (\beta < \alpha \rightarrow \neg \beta \in a)))$*;*

2. Axiom of infinity (INF) *(existence of a limit ordinal):*
   $\exists \alpha (\exists \beta \beta < \alpha \wedge \forall \beta (\beta < \alpha \rightarrow \exists \gamma (\beta < \gamma \wedge \gamma < \alpha)))$;

3. Axiom of extensionality (EXT)*:* $\forall a, b (\forall \alpha (\alpha \in a \leftrightarrow \alpha \in b) \rightarrow a = b)$;

4. Initial segment axiom (INI)*:* $\forall \alpha \exists a \forall \beta (\beta < \alpha \leftrightarrow \beta \in a)$*;*

5. Boundedness axiom (BOU)*:* $\forall a \exists \alpha \forall \beta (\beta \in a \rightarrow \beta < \alpha)$*;*

6. Pairing axiom (GPF) *(Gödel Pairing Function):*
$\forall\alpha,\beta,\gamma\,(g(\beta,\gamma)\leq\alpha\leftrightarrow\ \forall\delta,\epsilon((\delta,\epsilon)<^*(\beta,\gamma)\ \rightarrow g(\delta,\epsilon)<\alpha))$.
Here $(\alpha,\beta)<^*(\gamma,\delta)$ *stands for*
$\exists\eta,\theta(\eta=\max(\alpha,\beta)\wedge\theta=\max(\gamma,\delta)\wedge(\eta<\theta\vee$
$(\eta=\theta\wedge\alpha<\gamma)\vee(\eta=\theta\wedge\alpha=\gamma\wedge\beta<\delta)))$,
where $\gamma=\max(\alpha,\beta)$ *abbreviates* $(\alpha>\beta\wedge\gamma=\alpha)\vee(\alpha\leq\beta\wedge\gamma=\beta)$;

7. Surjectivity of pairing (SUR)*:* $\forall\alpha\exists\beta,\gamma\,(\alpha=g(\beta,\gamma))$;

8. Axiom schema of separation (SEP)*: For all L_{SO}-formulae $\phi(\alpha,P_1,...,P_n)$ postulate:*
$\forall P_1,...,P_n\forall a\exists b\forall\alpha\,(\alpha\in b\leftrightarrow\alpha\in a\wedge\phi(\alpha,P_1,...,P_n))$;

9. Axiom schema of replacement (REP)*: For all L_{SO}-formulae $\phi(\alpha,\beta,P_1,...,P_n)$ postulate:*
$\forall P_1,...,P_n\,(\forall\xi,\zeta_1,\zeta_2(\phi(\xi,\zeta_1,P_1,...,P_n)\wedge\phi(\xi,\zeta_2,P_1,...,P_n)\rightarrow\zeta_1=\zeta_2)\rightarrow$
$\forall a\exists b\forall\zeta\,(\zeta\in b\leftrightarrow\exists\xi\in a\,\phi(\xi,\zeta,P_1,...,P_n)))$;

10. Powerset axiom (POW)*:*
$\forall a\exists b(\forall z(\exists\alpha\alpha\in z\wedge\forall\alpha(\alpha\in z\rightarrow\alpha\in a)\rightarrow\exists\xi\forall\beta(\beta\in z\leftrightarrow g(\beta,\xi)\in b)))$.

Theorem 26. *The theory* SO *can be interpreted in the theory* ZFC*.*

Definition 27. *For sets or classes* R, X, Y, f *define the following notions in* SO*:*

$$\begin{aligned}
\emptyset &:= \iota_0 = \{\alpha \,|\, \alpha \neq \alpha\} \\
\mathrm{dom}(R) &:= \{\alpha | \exists \beta ((\alpha, \beta) \in R)\} \\
\mathrm{ran}(R) &:= \{\beta | \exists \alpha ((\alpha, \beta) \in R)\} \\
\mathrm{fun}(f) &:= \forall \alpha, \beta_1, \beta_2 ((\alpha, \beta_1) \in f \wedge (\alpha, \beta_2) \in f \rightarrow \beta_1 = \beta_2) \\
f \colon X \rightarrow Y &:= \mathrm{fun}(f) \wedge \mathrm{dom}(f) = X \wedge \mathrm{ran}(f) \subset Y \\
\alpha = f(\beta) &:= (\alpha, \beta) \in f \\
\alpha R \beta &:= (\alpha, \beta) \in R \\
X \times Y &:= \{\gamma | G_1(\gamma) \in X \wedge G_2(\gamma) \in Y\} \\
X \upharpoonright Y &:= \{(\alpha, \beta) \in X | \alpha \in Y\}
\end{aligned}$$

Theorem 28. (SO) *Let $\phi(\alpha, X_1, \ldots, X_n)$ be an L_{SO}-formula. Then for all $X_1, \ldots, X_n$,*

$$\forall\alpha((\forall\beta<\alpha\,\phi(\beta, X_1, \ldots, X_n)) \rightarrow \phi(\alpha, X_1, \ldots, X_n))$$

implies

$$\forall\alpha\,\phi(\alpha, X_1, \ldots, X_n)$$

Theorem 29. (SO) *Let $R\colon \mathrm{On}\times\mathrm{SOn}\rightarrow\mathrm{On}$ be a function defined by some formula $\phi(\alpha, f, \beta, X_1, \ldots, X_n)$. Then there exists a unique function $F\colon\mathrm{On}\rightarrow\mathrm{On}$ defined by a formula $\psi(\alpha, \beta, X_1, \ldots, X_n)$ such that*

$$\forall\alpha(F(\alpha)=R(\alpha, F\restriction\iota_\alpha)) \qquad (1)$$

Assembling sets along wellfounded relations

Definition 30. *An ordered pair $x = (x, R_x)$ is a* point *if R_x is a wellfounded relation on ordinals and $x \in \mathrm{dom}(R_x)$. Let $\mathbb{P}$ be the class of all points. Unless specified otherwise we use R_x to denote the wellfounded relation of the point x .*

Define recursively

$$I_x\colon \mathrm{dom}(R_x) \to V, \text{ by } I_x(u) = \{I_x(v) \mid v R_x u\}.$$

Then let $I(x) = I_x(x)$ be the *interpretation* of x.

Note that for points x and y

$$\begin{aligned} I_x(u) = I_y(v) \ \text{ iff } & \ \{I_x(u') \mid u' R_x u\} = \{I_x(v') \mid v' R_y v\} \\ \text{ iff } & \ (\forall u' R_x u \, \exists v' R_y v \, I_x(u') = I_y(v')) \wedge (\forall v' R_y v \, \exists u' R_x u \, I_x(u') = I_y(v')). \end{aligned}$$

Definition 31. *Define a relation* $\equiv$ *on points* $x = (x, R_x)$, $y = (y, R_y)$ *by induction on the product wellorder* $R_x \times R_y$*:*

$$(x, R_x) \equiv (y, R_y) \text{ iff } \forall u R_x x \exists v R_y y\ (u, R_x) \equiv (v, R_y) \wedge \forall v R_y y \exists u R_x x\ (u, R_x) \equiv (v, R_y).$$

Lemma 32. (SO) $\equiv$ *is an equivalence relation on* $\mathbb{P}$*.*

Definition 33. *Let* $x = (x, R_x)$ *and* $y = (y, R_y)$ *be points. Then set*

$$x \blacktriangleleft y \text{ iff } \exists v R_y y\ x \equiv (v, R_y).$$

Lemma 34. (SO) *The equivalence relation* $\equiv$ *is a congruence relation with respect to* $\blacktriangleleft$*, i.e.,*

$$x \blacktriangleleft y \wedge x \equiv x' \wedge y \equiv y' \rightarrow x' \blacktriangleleft y'.$$

The class of points satisfies ZFC

Lemma 35. (SO) *Let $(x_i \mid i \in A)$ be a set-sized sequence of points. Then there is a point $y = (y, R_y)$ such that for all points x holds*

$$x \blacktriangleleft y \text{ iff } \exists i \in A\, x \equiv x_i\,.$$

Theorem 36. (SO) $\mathbb{P} = (\mathbb{P}, \equiv, \blacktriangleleft)$ *is a model of* ZFC.

Proof. (1) The axiom of *extensionality* holds in $\mathbb{P}$:

$$\forall x \forall y (\forall z (z \blacktriangleleft x \leftrightarrow z \blacktriangleleft y) \rightarrow x \equiv y).$$

(2) The axiom of *pairing* holds in $\mathbb{P}$:

$$\forall x \forall y \exists z \forall w (w \blacktriangleleft z \leftrightarrow (w \equiv x \vee w \equiv y)).$$

(3) The axiom of *unions* holds in $\mathbb{P}$:

$$\forall x \exists y \forall z (z \blacktriangleleft y \leftrightarrow \exists w (w \blacktriangleleft x \wedge z \blacktriangleleft w)).$$

(4) The *replacement schema* holds in $\mathbb{P}$, i.e., for every first-order formula $\varphi(u, v)$ in the language of $\equiv$ and $\blacktriangleleft$ the following is true in $\mathbb{P}$:

$$\forall u, v, v' ((\varphi(u, v) \wedge \varphi(u, v')) \rightarrow v \equiv v') \rightarrow \forall x \exists y \forall z (z \blacktriangleleft y \leftrightarrow \exists u (u \blacktriangleleft x \wedge \varphi(u, z))).$$

The replacement schema also implies the *separation* schema.

(5) The axiom of *powersets* holds in $\mathbb{P}$:

$$\forall x \exists y \forall z (z \blacktriangleleft y \leftrightarrow \forall w (w \blacktriangleleft z \rightarrow w \blacktriangleleft x)).$$

(6) The *axiom of choice* holds in $\mathbb{P}$:

$$\forall x ((\forall y, z ((y \blacktriangleleft x \wedge z \blacktriangleleft x) \rightarrow (\exists u u \blacktriangleleft y \wedge (\neg y \equiv z \rightarrow \neg \exists u (u \blacktriangleleft y \wedge u \blacktriangleleft z))))) \rightarrow$$
$$\exists w \forall y (y \blacktriangleleft x \rightarrow \exists u ((u \blacktriangleleft w \wedge u \blacktriangleleft y) \wedge \forall v ((v \blacktriangleleft w \wedge v \blacktriangleleft y) \rightarrow u \equiv v)))).$$

qed(6)

(7) The *foundation schema* holds in $\mathbb{P}$, i.e., for every first-order formula $\varphi(u)$ in the language of $\equiv$ and $\blacktriangleleft$ the following is true in $\mathbb{P}$:

$$\exists u \varphi(u) \rightarrow \exists y (\varphi(y) \wedge \forall z (z \blacktriangleleft y \rightarrow \neg \varphi(z)).$$

(8) The axiom of *infinity* holds in $\mathbb{P}$, i.e.,

$$\exists x (\exists y \, y \blacktriangleleft x \wedge \forall y (y \blacktriangleleft x \rightarrow \exists z (z \blacktriangleleft x \wedge \forall u (u \blacktriangleleft z \leftrightarrow u \blacktriangleleft y \vee u \equiv y))))$$

$\square$

Theorem 37. *In the set theoretical universe V consider a class $\mathcal{S} \subseteq \{x \mid x \subseteq \mathrm{Ord}\}$ such that $\mathcal{S} = (\mathrm{Ord}, \mathcal{S}, <, =, \in, G)$ is a model of the theory* SO. *Then there is a unique inner model $(M, \in)$ of* ZFC *such that $\mathcal{S} = \{v \in M \mid v \subseteq \mathrm{Ord}\}$.*

3-adic representations and ordinal stacks

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

Proposition 38. *Let* >1 *be a fixed* basis *ordinal. An equality*

$$\alpha = \delta^{\alpha_0} \cdot \zeta_0 + \delta^{\alpha_1} \cdot \zeta_1 + \ldots + \delta^{\alpha_{n-1}} \cdot \zeta_{n-1}$$

with $\alpha_0 > \alpha_1 > \ldots > \alpha_{n-1}$ *and* $0 < \zeta_0, \zeta_1, \ldots, \zeta_{n-1} < \delta$ *is called a* δ-adic representation *of* α*. We claim that every* $\alpha \in \mathrm{Ord}$ *possesses a unique* δ*-adic representation.*

By the proposition, a decreasing stack $\alpha_0 > \alpha_1 > \ldots > \alpha_{n-2} \geqslant \alpha_{n-1}$ of ordinals can be coded by one ordinal

$$\alpha = \langle \alpha_0, \alpha_1, \ldots, \alpha_{n-2}, \alpha_{n-1} \rangle = 3^{\alpha_0} + 3^{\alpha_1} + \ldots + 3^{\alpha_{n-2}} + 3^{\alpha_{n-1}}.$$

Proposition 39. *Let $t \in \mathrm{Ord}$ be a limit time and $t_0 < t$. For time $\tau \in [t_0, t)$ let the contents of the register* `stack` *be of the form $\alpha_\tau = \langle \alpha_0, \ldots, \alpha_{k-1}, \rho(\tau), \ldots \rangle$ for fixed $\alpha_0, \ldots, \alpha_{k-1}$ and variable $\rho(\tau) \leqslant \alpha_{k-1}$. Assume that the sequence $(\rho(\tau) \mid \tau \in [t_0, t))$ is weakly monotonously increasing and that the length of* `stack` *is equal to $k+1$ cofinally often below t. Then at limit time t the content of* `stack` *is of the form $\alpha_t = \langle \alpha_0, \ldots, \alpha_{k-1}, \rho \rangle$ with $\rho = \bigcup_{\tau \in [t_0, t)} \rho(\tau)$.*

Stack recursion

```
value:=2                %% set value to undefined
MainLoop:
  nu:=last(stack)
  alpha:=llast(stack)
  if nu = alpha then
1:  do
    remove_last_element_of(stack)
    value:=0            %% set value equal to 0
    goto SubLoop
    end_do
  else
2:  do
    stack:=stack + 1    %% push the ordinal 0 onto the stack
    goto MainLoop
    end_do
```

```
SubLoop:
  nu:=last(stack)
  alpha:=llast(stack)
  if alpha = UNDEFINED then STOP
  else
    do
    if H(alpha,nu,value)=1 then
3:    do
      remove_last_element_of(stack)
      value:=1
      goto SubLoop
      end_do
    else
4:    do
      stack:=stack + 2*(3**y)  %% push y+1
      value:=2             %% set value to undefined
      goto MainLoop
      end_do
    end_do
```

Theorem 40. *The above program P has the following properties*

a) *If P is in state* `MainLoop` *at time s with* `stack` *contents $\langle\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\rangle$ where $n \geqslant 1$ then it will get into state* `SubLoop` *at a later time t with the same* `stack` *contents $\langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}\rangle$ and the register* `value` *holding the value $F(\alpha_{n-1})$. Moreover in the time interval $[s, t)$ the contents of* `stack` *will always be at least as big as $\langle\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\rangle$.*

b) *Let P be in state* `MainLoop` *at time s with* `stack` *contents $\alpha_0 > \alpha_1 > \alpha_2 > \ldots > \alpha_{n-1}$ where $n \geqslant 1$. Define $\bar{\alpha}$ = the minimal ordinal $\nu < \alpha_{n-1}$ such that $H(\alpha_{n-1}, \nu, F(\nu)) = 1$ if this exists and $\bar{\alpha} = \alpha_{n-1}$ else. Then there is a strictly increasing sequence $(t_i | i \leqslant \bar{\alpha})$ of times $t_i > t$ such that P is in state* `MainLoop` *at time t_i with* `stack` *contents $\langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, i\rangle$, and in every time interval $\tau \in [t_i, t_{i+1})$ the* `stack` *contents are $\geqslant \langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, i\rangle$.*

c) *If P is in state* `MainLoop` *with* `stack` *contents α then it will later* stop *with* `stack` *content α and the register* `value` *holding the value $F(\alpha)$. Hence the function F is ordinal register computable.*

A recursive truth predicate

Consider a language L_R appropriate for first-order structures of the form

$$(\alpha, <, G, R).$$

The language has atomic formulas $t_1 \equiv t_2$, $t_1 < t_2$, $\dot{G}(t_1, t_2, t_3)$ and $\dot{R}(t_1)$. If φ and ψ are (compound) formulas of the language and t is a term then

$$\neg\varphi,\ (\varphi \vee \psi),\text{ and } \exists v_n < t\ \varphi$$

are also formulas.

We arrange a coding by ordinals such that a bounded existential quantification $\exists v_n < c_\xi\, \varphi$ is coded by a larger ordinal than each of its instances $\varphi\frac{c_\zeta}{v_n}$ with $\zeta < \xi$:

$$\varphi\frac{c_\zeta}{v_n} < (\exists v_n < c_\xi\, \varphi).$$

The satisfaction relation

$$(\alpha, <, G, R) \vDash \varphi$$

This is equivalent to

$$(\varphi, <, G, R) \vDash \varphi.$$

Definition 41. *Define a* bounded truth predicate $T \subseteq \mathrm{Ord}$ *over the ordinals recursively by*

$$T(\alpha) \quad \textit{iff} \quad \alpha \textit{ is a bounded } L_T\textit{-sentence and } (\alpha, <, G, T \cap \alpha) \vDash \alpha.$$

The characteristic function χ_T of T can be defined according to the recursion scheme

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

where

$$\begin{array}{rll} H(\alpha, \nu, \chi) = 1 & \text{iff} & \alpha \text{ is an } L_T\text{-sentence and} \\ & & \big(\exists \xi, \zeta < \alpha\, (\alpha = c_\xi \equiv c_\zeta \wedge \xi = \zeta) \\ & \text{or} & \exists \xi, \zeta < \alpha\, (\alpha = c_\xi < c_\zeta \wedge \xi < \zeta) \\ & \text{or} & \exists \xi, \zeta, \eta < \alpha\, (\alpha = \dot{G}(c_\xi, c_\zeta, c_\eta) \wedge \eta = G(\xi, \zeta)) \\ & \text{or} & \exists \xi < \alpha\, (\alpha = \dot{R}(c_\xi) \wedge \nu = \xi \wedge \chi = 1) \\ & \text{or} & \exists \varphi < \alpha\, (\alpha = \neg\varphi \wedge \nu = \varphi \wedge \chi = 0) \\ & \text{or} & \exists \varphi, \psi < \alpha\, (\alpha = (\varphi \vee \psi) \wedge (\nu = \varphi \vee \nu = \psi) \wedge \chi = 1) \\ & \text{or} & \exists n < \omega \exists \xi < \alpha \exists \varphi < \alpha\, (\alpha = \exists v_n < c_\xi\, \varphi \wedge \exists \zeta < \xi\; \nu = \varphi \frac{c_\zeta}{v_n} \wedge \chi = 1)\big). \end{array}$$

H and thus the bounded truth predicate T are ordinal computable.

Computing a model of set theory

Definition 42. *For ordinals μ and α define*

$$T(\mu,\alpha)=\{\beta<\mu\,|\,T(G(\alpha,\beta))=1\}.$$

Set

$$\mathcal{S}=\{T(\mu,\alpha)\,|\,\mu,\alpha\in\mathrm{Ord}\}.$$

Theorem 43. $(\mathrm{Ord}, \mathcal{S}, <, =, \in, G)$ *is a model of the theory* SO.

Proof. Of axiom schemas (8) and (9):

For $\theta \in \mathrm{Ord}$ define

$$\mathcal{S}_\theta = \{T(\mu, \alpha) \mid \mu, \alpha \in \theta\}.$$

Then for any L_{SO}-formula $\varphi(v_0, \ldots, v_{n-1})$ and $\eta \in \mathrm{Ord}$ there is some limit ordinal $\theta > \eta$ such that

$$\forall \xi_0, \ldots, \xi_{n-1} \in \theta\, ((\mathrm{Ord}, \mathcal{S}, <, =, \in, G) \vDash \varphi[\xi_0, \ldots, \xi_{n-1}] \text{ iff } (\theta, \mathcal{S}_\theta, <, =, \in, G) \vDash \varphi[\xi_0, \ldots, \xi_{n-1}]).$$

$$\forall \xi_0, \ldots, \xi_{n-1} \in \theta\, ((\mathrm{Ord}, \mathcal{S}, <, =, \in, G) \vDash \varphi[\xi_0, \ldots, \xi_{n-1}] \text{ iff } (\theta, <, G \cap \theta^3, T) \vDash \varphi^*[\xi_0, \ldots, \xi_{n-1}]).$$

So sets witnessing axioms (8) and (9) can be defined over $(\theta, <, G \cap \theta^3, T)$ and are thus elements of $\mathcal{S}$. $\square$

Ordinal computability corresponds to constructibility

Kurt Gödel: The inner model L of *constructible sets*

$$L = \bigcup_{\alpha \in \mathrm{Ord}} L_\alpha$$

where $L_0 = \emptyset$, $L_\delta = \bigcup_{\alpha<\delta} L_\alpha$ for limit ordinals δ, and $L_{\alpha+1} =$ the set of all sets which are first-order definable in the structure $(L_\alpha, \in)$.

The model L is the $\subseteq$-smallest inner model of set theory.

Theorem 44. *A set x of ordinals is ordinal computable if and only if it is an element of the constructible universe L.*

Proof. Let $x \subseteq \mathrm{Ord}$ be ordinal computable by the program P from the ordinals $\delta_1, \ldots, \delta_{n-1}$, so that for every $\alpha \in \mathrm{Ord}$:

$$P \colon (\alpha, \delta_1, \ldots, \delta_{n-1}, 0, 0, \ldots) \mapsto \chi_x(\alpha).$$

By the simple nature of the computation procedure the same computation can be carried out inside the inner model L, so that for every $\alpha \in \mathrm{Ord}$:

$$(L, \in) \vDash P \colon (\alpha, \delta_1, \ldots, \delta_{n-1}, 0, 0, \ldots) \mapsto \chi_x(\alpha).$$

Hence $\chi_x \in L$ and $x \in L$.

Conversely consider $x \in L$. Since $(\mathrm{Ord}, \mathcal{S}, <, =, \in, G)$ is a model of the theory SO there is an inner model M of set theory such that

$$\mathcal{S} = \{z \subseteq \mathrm{Ord} \mid z \in M\}.$$

Since L is the $\subseteq$-smallest inner model, $L \subseteq M$. Hence $x \in M$ and $x \in \mathcal{S}$. Let $x = T(\mu, \alpha)$. By the computability of the truth predicate, x is ordinal register computable from the parameters μ and α. $\square$

An application: the generalised continuum hypothesis in L

Theorem 45. *The constructible model $(L, \in)$ satisfies that every set of ordinals is ordinal computable.*

Proof. Let $x \in L$, $x \subseteq \mathrm{Ord}$, let P be a program and $\delta_1, \ldots, \delta_{n-1} \in \mathrm{Ord}$ such that for every $\alpha \in \mathrm{Ord}$:

$$P\colon (\alpha, \delta_1, \ldots, \delta_{n-1}, 0, 0, \ldots) \mapsto \chi_x(\alpha).$$

The same computation can be carried out inside the inner model L:

$$(L, \in) \vDash P\colon (\alpha, \delta_1, \ldots, \delta_{n-1}, 0, 0, \ldots) \mapsto \chi_x(\alpha).$$

So in L, x is ordinal computable. $\square$

Theorem 46. *Assume that every set of ordinals is ordinal computable. Then:*

a) *Let $\kappa \geqslant \omega$ be an infinite ordinal and $x \subseteq \kappa$. Then there are ordinals $\alpha_1, \ldots, \alpha_{n-1} < \kappa^+$ such that x is ordinal computable from the parameters $\alpha_1, \ldots, \alpha_{n-1}$.*

b) *Let $\kappa \geqslant \omega$ be infinite. Then $\mathrm{card}(\mathcal{P}(\kappa)) = \kappa^+$.*

c) *The generalised continuum hypothesis* GCH *holds.*

Proof. a) Take a program P and $\alpha'_1, \ldots, \alpha'_{n-1} \in \mathrm{Ord}$ such that for every $\alpha \in \mathrm{Ord}$:

$$P\colon (\alpha, \alpha'_1, \ldots, \alpha'_{n-1}, 0, 0, \ldots) \mapsto \chi_x(\alpha).$$

Let θ be an upper bound for the lengths of these computations for $\alpha < \kappa$. Take a transitive ZF^--model $(M, \in)$ such that $\alpha'_1, \ldots, \alpha'_{n-1}, \theta, \kappa, x \in M$. Since ordinal computations are *absolute* for models of set theory, for all $\alpha < \kappa$:

$$(M, \in) \vDash P\colon (\alpha, \alpha'_1, \ldots, \alpha'_{n-1}, 0, 0, \ldots) \mapsto \chi_x(\alpha).$$

The downward LÖWENHEIM-SKOLEM theorem and the MOSTOWSKI isomorphism theorem yield an elementary embedding

$$\pi\colon(\bar{M},\in)\to(M,\in)$$

such that $\bar{M}$ is transitive, $\operatorname{card}(\bar{M})=\kappa$ and $\{\alpha_1', \ldots, \alpha_{n-1}', \theta, \kappa, x\}\cup\kappa\subseteq\pi''\bar{M}$. Let $\pi(\alpha_1)=\alpha_1'$, ..., $\pi(\alpha_{n-1}')=\alpha_{n-1}$. Then $\alpha_1, \ldots, \alpha_{n-1}<\kappa^+$ since $\operatorname{card}(\bar{M})<\kappa^+$. Observe that $\pi(x)=x$. Since π is elementary $(\bar{M},\in)$ satisfies for $\alpha<\kappa$ that

$$(\bar{M},\in)\vDash P\colon(\alpha,\alpha_1,\ldots,\alpha_{n-1},0,0,\ldots)\mapsto\chi_x(\alpha).$$

By the absoluteness of ordinal computations between $\bar{M}$ and V

$$P\colon(\alpha,\alpha_1,\ldots,\alpha_{n-1},0,0,\ldots)\mapsto\chi_x(\alpha)$$

for $\alpha<\kappa$. Thus x is ordinal computable from the parameters $\alpha_1,\ldots,\alpha_{n-1}<\kappa^+$. $\square$

Bibliography

[1] Ryan Bissell-Siders. Ordinal computers. Eprint at: arXiv:math.LO/9804076, 1998.

[2] Nigel J. Cutland. *Computability: An introduction to Recursive Function Theory*. Perspectives in Mathematical Logic. Cambridge University Press, 1980.

[3] Keith Devlin. *Constructibility*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1984.

[4] Kurt Gödel. *The Consistency of the Continuum Hypothesis*, volume 3 of *Ann. of Math. Studies*. Princeton University Press, Princeton, 1940.

[5] Joel David Hamkins and Andy Lewis. Infinite Time Turing Machines. *J. Symbolic Logic*, 65(2):567–604, 2000.

[6] Thomas Jech. *Set Theory. The Third Millennium Edition*. Springer Monographs in Mathematics. Springer-Verlag, 2003.

[7] Peter Koepke. Turing computations on ordinals. *Bull. Symbolic Logic*, 11(3):377–397, 2005.