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.

1 Introduction.

There are many equivalent machine models for defining the class of intuitively computable sets. We define computations on ordinals in analogy to the *unlimited register machines* (*URM*) presented in [2]. An URM has registers $R_0, R_1, \ldots$ which can hold *natural numbers*, i.e., elements of the set $\omega = \{0, 1, \ldots\}$. A register program consists of commands to increase or to reset a register. The program may loop on condition of equality between two registers. A natural generalisation from the perspective of transfinite ordinal theory is to extend such calculations to the class $\mathrm{Ord} = \{0, 1, \ldots, \omega, \omega + 1, \ldots\}$ of all *ordinal numbers* so that registers may contain arbitrary ordinals. At *limit* ordinals one defines the program states and the registers contents by appropriate limit operations which may be viewed as *inferior limits* (liminf).

This notion of *ordinal (register) computability* obviously extends standard register computability. By the Church-Turing thesis many operations on natural numbers are ordinal computable. The ordinal arithmetic operations (addition, multiplication, exponentiation) and Gödel's pairing function $G\colon \mathrm{Ord} \times \mathrm{Ord} \to \mathrm{Ord}$ are also ordinal computable.

Using the pairing function one can view each ordinal α as a first-order sentence with constant symbols for ordinals $< \alpha$. One can then define a recursive truth predicate $T \subseteq \mathrm{Ord}$ by:

$$\alpha \in T \text{ iff } (\alpha, <, G \cap \alpha^3, T \cap \alpha) \vDash \alpha.$$

This recursion can be carried out on an ordinal register machine, using stacks which contain finite decreasing sequences of ordinals. For ordinals μ and ν the predicate T codes the set

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

The class

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

is the class of sets of ordinals of a transitive proper class model of set theory. Since the ordinal computations can be carried out in the $\subseteq$-smallest such model, namely Gödel's model L of *constructible sets*, we obtain the main result characterising ordinal computability:

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

This theorem may be viewed as an analogue of the Church-Turing thesis: ordinal computability defines a natural and absolute class of sets, and it is stable with respect to technical variations in its definition. Register machines on ordinals were first considered by Ryan Bissell-Siders [1]; the results proved in this article were guided by the related theory of ordinal Turing machines [7] which generalises the infinite-time Turing machines of [5].

There are several open questions and projects connected with ordinal computability:

- 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?

Our tutorial on ordinal computations will be structured as follows:

- 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.

2 Ordinal numbers

Set theory, naively or axiomatically, is a natural, strong and convenient theory which can be used as a foundation for all of mathematics. I.e., the standard notions can be defined naturally and their usual properties can be shown naively or from the axioms.

Numbers, in particular natural and real numbers, are the most important mathematical notions. The real numbers can be obtained from the natural numbers in the usual set theoretic way: natural numbers $\rightarrow$ rational number $\rightarrow$ Dedekind cuts $\equiv$ real numbers. In set theory one considers an infinitary extension of the natural numbers: Cantor's *ordinal numbers* allow to count beyond the natural numbers into the *transfinite*. We shall use a formalisation which is usually associated with JOHN VON NEUMANN. We motivate the formalisation by introducing (some) natural numbers in a seemingly ad hoc way.

In a sense which can be made precise, sets can iteratively be generated from the empty set $\emptyset$. We also base numbers on the empty set. Define recursively

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

Obviously, the n-th set n has exactly n elements and we have chosen adequate representatives for the intuitively given "standard" numbers. We state some facts about our numbers which will lead to a general definition of number.

Proposition 2. *Let m,n be numbers as above. Then $m\in n$ iff the corresponding standard numbers satisfy $m<n$. So for the above numbers, $\in$ is isomorphic to the standard ordering $<$ on natural numbers.*

2.1 Definitions

Definition 3. *A set or class A is* transitive*, $\mathrm{Trans}(A)$, iff $\forall u, v(u \in v \wedge v \in A \rightarrow u \in A)$.*

Obviously:

Proposition 4. *Let m be a number as in (1). Then*

a) m is transitive;

b) every element of m is transitive.

This leads to

Definition 5. *A set x is an* ordinal number*, $\mathrm{Ord}(x)$, if $\mathrm{Trans}(x) \wedge \forall y \in x\, \mathrm{Trans}(y)$. Let*

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

be the class of all ordinals.

The class Ord contains the above natural numbers. We use small greek letters α, β, γ, ... as variables ranging over ordinals. We write $\alpha < \beta$ instead of $\alpha \in \beta$ and $\alpha \leqslant \beta$ instead of $\alpha < \beta \vee \alpha = \beta$. Under appropriate set-theoretic axioms the class Ord is strongly well-ordered by the relation $<$. Let us recall the *axiom of foundation* which asserts the existence of $\in$-minimal elements of sets:

$$\forall x(\exists y y \in x \rightarrow \exists y(y \in x \wedge \neg \exists z(z \in x \wedge z \in y))).$$

Theorem 6. *a) The class* Ord *is transitive.*

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

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

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

Proof. a) Let $x \in \alpha \in \mathrm{Ord}$. Since α is an ordinal we have $\mathrm{Trans}(x)$. Consider $y \in x$. Since α is transitive we have $x \in \alpha$ and so $\mathrm{Trans}(x)$. Thus $\forall y \in x\, \mathrm{Trans}(y)$ and $x \in \mathrm{Ord}$.

b) Let $\alpha, \beta, \gamma \in \mathrm{Ord}$ and $\alpha < \beta < \gamma$. Then $\alpha < \gamma$ by the transitivity of the ordinal γ.

Let $\alpha \in \mathrm{Ord}$. Then $\alpha \notin \alpha$ and so $\alpha \not< \alpha$.

For the linearity of $<$ assume that there are ordinals α, β such that

$$\alpha \not< \beta,\ \alpha \neq \beta, \text{ and } \beta \not< \alpha.$$

By the axiom of foundation we can assume that α is minimal with that property, and that with respect to α the ordinal β is minimal with that property. We claim that $\alpha = \beta$.

Let $\xi \in \alpha$. By the minimality of α we have $\xi < \beta$ or $\xi = \beta$ or $\beta < \xi$. Assume $\xi = \beta$ or $\beta < \xi$. Then $\beta < \alpha$ contradicting the minimal choice of α. Hence $\xi \in \beta$.

Conversely let $\xi \in \beta$. By the minimality of β we have $\xi < \alpha$ or $\xi = \alpha$ or $\alpha < \xi$. Assume $\xi = \alpha$. Then $\alpha < \beta$, contradicting the choice of α and β. Assume $\alpha < \xi$. Then again $\alpha < \beta$, contradiction. Thus $\xi \in \alpha$.

But $\alpha = \beta$ contradicts the choice of α and β.

c) follows directly from the axiom of foundation. □

There are further important structural properties of ordinals:

Exercise 1. a) For $\alpha, \beta \in \mathrm{Ord}$ we have $\alpha \leqslant \beta$ iff $\alpha \subseteq \beta$. b) If $x \subseteq \mathrm{Ord}$ is a set of ordinals then $\bigcup x \in \mathrm{Ord}$ and $\bigcap x \in \mathrm{Ord}$. c) For every $\alpha \in \mathrm{Ord}$ the set $\alpha + 1 = \alpha \cup \{\alpha\}$ is the immediate successor of α with respect to $<$.

Definition 7. *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 $\alpha \neq 0$.*

The *axiom of infinity* states that there exists a limit ordinal.

Definition 8. *Let ω be the smallest limit ordinal. A set n is a* natural number *if $n < \omega$. So ω is the set of* natural numbers*.*

The latter definition is justified by

Theorem 9. *The structure $(\omega, 0, +1)$ satisfies the* PEANO *axioms. In particular the principle of* complete induction *holds:*

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

2.2 Induction and recursion

The natural numbers form an initial segment of the ordinal numbers:

$$n \in \omega \rightarrow n \in \mathrm{Ord}.$$

The most remarkable fact about ordinals is that the principles of induction and recursion can be extended from the natural numbers to all the ordinals.

Theorem 10. *Let $\varphi(v, \vec{w})$ be an $\in$-formula. Then*

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

Proof. Assume $\varphi(\alpha, \vec{w})$. Let $x = \{\beta \leqslant \alpha \mid \varphi(\beta, \vec{w})\}$. Then $x \neq \emptyset$. By the axiom of foundation take an $\in$-minimal element $\alpha' \in x$. Then $\forall \beta < \alpha' \neg \varphi(\beta, \vec{w})$ and

$$\varphi(\alpha', \vec{w}) \wedge \forall \beta < \alpha' \neg \varphi(\beta, \vec{w})). \qquad \square$$

This theorem can be reformulated as an *induction principle* which looks more like the familiar principle of complete induction. According to the various types of ordinals one distinguishes the *initial* case 0, the *successor* case, and the *limit* case.

Theorem 11. *Let $\varphi(v, \vec{w})$ be an $\in$-formula and assume that*

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

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

The most important transfinite construction principle is construction by *recursion* along the ordinals.

Theorem 12. *Let $G: V \rightarrow V$ 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*

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

Proof. The function F may be defined as the union of all set-sized *approximations* behaving similarly:

$$F = \bigcup \{f \mid \exists \gamma \in \mathrm{Ord} \, (f: \gamma \rightarrow V \wedge \forall \alpha \in \gamma f(\alpha) = G(f \upharpoonright \alpha))\}.$$

Using the axioms of *replacement* and *union* the existence of sufficiently many compatible approximations f can be shown by ordinal induction. $\square$

The *recursion rule* G will usually be described separately for successor and limit ordinals and the initial case 0:

Theorem 13. *Let $G_0 \in V$ and let $G_{\text{succ}}: V \to V$ and $G_{\lim}: V \to V$ be definable functions. Then there is a unique definable function $F: \text{Ord} \to V$ such that*

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

Exercise 2. Prove this form of the recursion theorem from the recursion theorem 12.

An example of a recursive construction is the VON NEUMANN hierarchy $(V_\alpha \mid \alpha \in \text{Ord})$ with

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

Here $\mathcal{P}(.)$ denotes the *power set* operation which by the *powerset axiom* maps sets to sets.

2.3 Ordinal arithmetic

The standard arithmetic operations have well-known recursive definitions which can be extended to all the ordinals by transfinite recursion.

Definition 14. *The* ordinal sum $\alpha + \beta$ *is defined by recursion on* $\beta \in \text{Ord}$ *by*

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

Definition 15. *The* ordinal product $\alpha \cdot \beta$ *is defined by recursion on* $\beta \in \text{Ord}$ *by*

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

These operations obviously extend the arithmetic on natural numbers.

Exercise 3. Exhibit explicit recursion rules for $+$ and $\cdot$ as in the recursion theorem 12.

Exercise 4. Prove the following arithmetic laws for ordinal arithmetic:

a) $(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$.

b) $(\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)$.

c) $\alpha \cdot (\beta + \gamma) = \alpha \cdot \beta + \alpha \cdot \gamma$.

Show that the operations are *not* commutative, and that the distributive law $(\alpha + \beta) \cdot \gamma = \alpha \cdot \gamma + \alpha \cdot \gamma$ fails.

Exercise 5. Prove

a) $\forall \alpha \exists \beta\, \alpha + \beta = \beta$.

b) $\forall \alpha \exists \beta\, \alpha \cdot \beta = \beta$.

The operations $+$ and $\cdot$ are *continuous* at limit ordinals with respect to *ordinal limits*:

Definition 16. *Let $(\delta_i \mid i < \lambda)$ 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* $(\delta_i \mid i < \lambda)$;

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

Exercise 6. Define a topology on the class Ord such that limit ordinals are limit points in the sense of the topology and such that the operations $+$ and $\cdot$ are continuous in the sense of the topology.

2.4 The GÖDEL pairing function

Definition 17. *Define a well-ordering* $\prec$ *on* Ord × Ord *by*

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

Exercise 7. Show that $\prec$ is a set-like well-ordering of Ord × Ord. *Set-like* means that

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

Definition 18. *Define a function* G^{-1}: Ord → Ord × Ord *recursively by*

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

Theorem 19. $G^{-1}\colon(\mathrm{Ord},<)\to(\mathrm{Ord}\times\mathrm{Ord},\prec)$ *is an order-isomorphism.*

Proof. $G^{-1}(\alpha)$ is defined for every $\alpha\in\mathrm{Ord}$ since $\prec$ is set-like and so

$$\mathrm{Ord}\times\mathrm{Ord}\setminus\{G^{-1}(\beta)\,|\,\beta<\alpha\}\neq\emptyset.$$

The definition of $G^{-1}(\alpha)$ immediately implies that G^{-1} is injective. For the surjectivity assume the contrary and let (γ',δ') be $\prec$-minimal such that $(\gamma',\delta')\notin\mathrm{range}(G^{-1})$. $\{(\gamma,\delta)\,|\,(\gamma,\delta)\prec(\gamma',\delta')\}$ is a set. By the replacement axiom choose an ordinal α such that

$$\forall(\gamma,\delta)\prec(\gamma',\delta')\,\exists\beta<\alpha\,G^{-1}(\beta)=(\gamma,\delta).$$

But then the recursive definition of G^{-1} will imply that $G^{-1}(\alpha)=(\gamma',\delta')$. Contradiction.

The recursive definition also implies directly that

$$\beta<\alpha\leftrightarrow G^{-1}(\beta)\prec G^{-1}(\alpha).$$

□

Exercise 8. Compute $G^{-1}(n)$ for $n=1,\ldots,6$. What is $G^{-1}(\omega)$?

Definition 20. *Let* G *be the inverse of the function* G^{-1}. G: Ord × Ord ↔ Ord *is called the* GÖDEL *pairing function for ordinals. Let* G_0 *and* G_1 *the components of* G^{-1}, *i.e.,*

$$\forall\alpha\,G(G_0(\alpha),G_1(\alpha))=\alpha.$$

3 Register machines

There are many equivalent machine models for defining the class of intuitively computable sets. We base our presentation on the *unlimited register machine* presented in [2].

3.1 Unlimited register machines - URMs

Definition 21. *An* unlimited register machine *URM has registers* R_0, R_1, ... *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* $R(t)\colon \omega \rightarrow \omega$. $R(t)(n)$ *is the content of the register* R_n *at time t. We also write* $R_n(t)$ *instead of* $R(t)(n)$.

Definition 22. *Let* $P = I_0, I_1, \ldots, I_{s-1}$ *be a program. A triple*

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

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*

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

   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*

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

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

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

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 imput $R(0)$. *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* $P\colon R(0) \mapsto R(\beta)(0)$.

3.2 Algorithms

It can be shown that the unlimited register machine is equivalent to the other standard models of computations like TURING machines. So a function $f\colon \omega \rightarrow \omega$ is computable by the URM iff it is TURING computable. In view of our later generalisations we present some arithmetic register programs:

```
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
```

Exercise 9. Write the addition program in the form $P = I_0, I_1, \ldots, I_{s-1}$ as in Definition 21.

Observe that the function $n \mapsto n \dot{-} 1$ is not a basic function of the URM. It can, however, be programmed as follows:

```
Decrementation, computing beta = alpha ∸ 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:
1  beta':=0
2  gamma:=0
3  if beta=beta' then STOP
4  beta':=beta'+1
5  gamma:=gamma + alpha
6  go to 3
```

Exercise 10. Write a program for division with remainder.

We interpret the program line `gamma:=gamma + alpha` as a *macro*, i.e., the above addition program has to be substituted for that line with reasonable modifications of variables, registers and line numbers. Also transfer of arguments and values between variables has to be arranged. This could, e.g., be achieved as follows:

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

```
17 go to 3
```

4 Ordinal computations

The URM is based on the operations $x := 0$ and $x := x + 1$ on natural numbers. An obvious generalisation from the perspective of transfinite ordinal theory is to extend such calculations to the class Ord of all *ordinal numbers* an let registers contain arbitrary ordinals. At *limit* ordinals one defines the program states and the registers contents by appropriate limit operations which may be viewed as *inferior limits*. Note that we shall use exactly the same programs for ordinal computations as for computations with natural numbers.

This notion of *ordinal (register) computability* obviously extends standard register computability. By the CHURCH-TURING thesis many operations on natural numbers are ordinal computable. The ordinal arithmetic operations (addition, multiplication, exponentiation) and GÖDEL's pairing function $G\colon \mathrm{Ord} \times \mathrm{Ord} \to \mathrm{Ord}$ are also ordinal computable.

4.1 Ordinal register machines - ORMs

Definition 23. *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 $R(t+1)\colon \omega \to \mathrm{Ord}$ 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 $R(t+1)\colon \omega \to \mathrm{Ord}$ 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 $R(t+1)\colon \omega \to \mathrm{Ord}$ by*

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

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

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

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

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

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* $P: R(0) \mapsto R(\beta)(0)$.

The definition of $I(t)$ for limit t can be motivated as follows. Since a program is finite its execution will lead to some (complex) looping structure involving loops, subloops and so forth. This can be presented by pseudo code like:

```
         ...
-->  17:begin loop
           ...
     21:   begin subloop
              ...
     29:   end subloop
           ...
     32:end loop
         ...
```

Assume that for times $r \rightarrow t$ 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

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

The interpretation of programs yields associated notions of computability.

Definition 24. *An* n*-ary partial function* F: $\mathrm{Ord}^n \rightharpoonup \mathrm{Ord}$ *is* (ordinal register) computable *if there is a register program* P *such that for every* n*-tuple* $(\alpha_0, \ldots, \alpha_{n-1})$ *holds*

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

Definition 25. *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*

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

where χ_x *is the characteristic function of* x.

4.2 Ordinal algorithms

Since ordinal register machines are a straightforward extension of standard register machines, all recursive functions can be computed by an ordinal register machine. The basic operations on ordinal numbers are ordinal register computable by the same URM programs that we used before:

```
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 -̇ 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
```

Note that by the liminf rule, at limit times t, the register contents will be $\alpha' = \beta = t$. The program computes the ordinal predecessor function

$$\alpha \dot{-} 1 = \begin{cases} \beta, & \text{if } \alpha = \beta + 1 \\ \alpha, & \text{else} \end{cases}$$

Ordinal computability is closed under compositions:

Theorem 26. *Let $f(v_0, ..., v_{n-1})$ and $g_0(\vec{w}), ..., g_{n-1}(\vec{w})$ be computable functions on the ordinals. Then the composition $h(\vec{w}) = f(g_0(\vec{w}), ..., g_{n-1}(\vec{w}))$ is ordinal register computable.*

The ordinal exponentiation function $\alpha \mapsto \beta^\alpha$ will be important for the sequel:

```
Ordinal exponentiation, computing gamma = beta ** alpha:
1  gamma:=1
2  alpha':=0
3  if alpha=alpha' then STOP
4  gamma=gamma * beta
5  alpha'=alpha'+1
6  go to 3
```

The GÖDEL pairing function is also ordinal computable:

```
Goedel pairing, computing gamma = G(alpha,beta):
0  alpha':=0
1  beta':=0
2  eta:=0
3  flag:=0
5  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
```

The inverse functions G_0 and G_1 satisfying

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

are ordinal computable as well. To compute $G_0(\gamma)$ compute $G(\alpha, \beta)$ for $\alpha, \beta < \gamma$ until you find α, β with $G(\alpha, \beta) = \gamma$; then set $G_0(\gamma) = \alpha$. This is a special case of the following inverse function theorem.

Theorem 27. *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*

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

5 The theory SO of sets of ordinals

Ordinal TURING computations do not directly produce highly hierarchical sets but ordinals and sets of ordinals. It is well-known that a model of ZERMELO-FRAENKEL set theory with the axiom of choice (ZFC) is determined by its sets of ordinals [6], Theorem 13.28. This motivates the formulation of a theory SO which axiomatises the sets of ordinals in a model of ZFC. The theory SO is two-sorted where the intended interpretations are *ordinals* and *sets of ordinals*. Let L_{SO} be the language

$$L_{\mathrm{SO}} := \{\mathrm{On}, \mathrm{SOn}, <, =, \in, g\}$$

where On and SOn are unary predicate symbols, $<$, $=$, and $\in$ are binary predicate symbols and g is a two-place function. The intended standard interpretation of g is given by the GÖDEL pairing function G. To simplify notation, we use lower case greek letters to range over elements of On and lower case roman letters to range over elements of SOn.

Definition 28. *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* $\phi(\alpha, P_1, \ldots, P_n)$ *postulate:*
   $\forall P_1, \ldots, P_n \forall a\exists b\forall\alpha\, (\alpha \in b \leftrightarrow \alpha \in a \wedge \phi(\alpha, P_1, \ldots, P_n))$*;*
9. Axiom schema of replacement (REP)*: For all* L_{SO}*-formulae* $\phi(\alpha, \beta, P_1, \ldots, P_n)$ *postulate:*
   $\forall P_1, \ldots, P_n\, (\forall\xi, \zeta_1, \zeta_2(\phi(\xi, \zeta_1, P_1, \ldots, P_n) \wedge \phi(\xi, \zeta_2, P_1, \ldots, P_n) \rightarrow \zeta_1 = \zeta_2) \rightarrow$
   $\forall a\exists b\forall\zeta\, (\zeta \in b \leftrightarrow \exists\xi \in a\, \phi(\xi, \zeta, P_1, \ldots, P_n)))$*;*
10. Powerset axiom (POW)*:*
   $\forall a\exists b(\forall z(\exists\alpha(\alpha \in z) \wedge \forall\alpha(\alpha \in z \rightarrow \alpha \in a) \rightarrow \exists\xi\forall\beta(\beta \in z \leftrightarrow g(\beta, \xi) \in b)))$.

In a model of ZFC the class of sets of ordinals together with the standard relations $<$, $=$, and $\in$, and the GÖDEL pairing function G constitutes a model of SO. Note that the powerset axiom of SO requires the axiom of choice since it stipulates the existence of *well-ordered* powersets. Thus:

Theorem 29. *The theory* SO *can be interpreted in the theory* ZFC.

For the converse direction, which will be proved in the two subsequent sections, we first indicate that all basic mathematical notions can be reasonably formalised within the system SO. Beyond the specific requirements of the present paper, this also shows that the theory SO might have some wider interest as a foundational theory.

For the formalisation of mathematics within SO we make use of the familiar class term notation $A = \{\alpha \,|\, \phi(\alpha)\}$ to denote *classes* of ordinals. If $A = \{\alpha \,|\, \phi(\alpha)\}$ is a non-empty class of ordinals let min (A) denote the *minimal element* of A. The existence of a unique *minimum* follows from the axioms (INI), (SEP) and (WO). (BOU) ensures the existence of an *upper bound* for each set a, the least of which will be noted lub(a). By (INI) the classes $\iota_\alpha := \{\beta|\beta < \alpha\}$ are sets. Using (SEP) and (INI), one sees that the *union* and *intersection* of two sets are again sets. Finite sets are denoted by $\{\alpha_0, \alpha_1, ..., \alpha_{n-1}\}$. Their existence is implied by (INI) and (SEP). We write POW(b, a) for b being a set satisfying (POW) for a; note that in SO the set b is *not* uniquely determined by a. ω denotes the least element of the class of limit numbers which by (INF) is not empty. Finally let $0 := \min(\{\alpha|\mathrm{On}(\alpha)\})$, $1 := \mathrm{lub}(\{0\})$, etc.

The inverse functions G_1, G_2 of G are defined via the properties $\alpha = G_1(\beta) \leftrightarrow \exists\gamma(\beta = G(\alpha, \gamma))$ resp. $\alpha = G_2(\beta) \leftrightarrow \exists\gamma(\beta = G(\gamma, \alpha))$. The axioms (GPF) and (SUR) imply the well-known properties of the GÖDEL pairing function and its projections, such as bijectivity and monotonicity properties. To simplify notation, write $(\alpha, \beta) := G(\alpha, \beta)$. Every set can be regarded as a set of pairs $a = \{(\alpha, \beta)|(\alpha, \beta) \in a\}$ or more generally as a set of n-tuples. In this way n-ary *relations* and *functions* on ordinals can be represented by sets. We define further notions connected with relations and functions.

Definition 30. *For sets or classes R, X, Y, f define the following notions in* SO*:*

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

The axioms of SO imply that these notions have their usual basic properties. We can now prove transfinite induction and recursion in SO.

Theorem 31. (SO) *Let $\phi(\alpha, X_1, ..., X_n)$ be an L_{SO}-formula. Then for all $X_1, ..., X_n$,*

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

implies

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

Proof. Otherwise, by (WO), there would be a minimal counterexample α, contradicting the assumption. □

Theorem 32. (SO) *Let R: On × SOn → On be a function defined by some formula $\phi(\alpha, f, \beta, X_1, ..., X_n)$. Then there exists a unique function F: On → On defined by a formula $\psi(\alpha, \beta, X_1, ..., X_n)$ such that*

$$\forall\alpha(F(\alpha) = R(\alpha, F\restriction\iota_\alpha)) \tag{2}$$

Proof. This is proved similar to the recursion theorem in ZF: We define the notion of *approximation functions* which are set-functions defined on proper initial segments of Ord, satisfying (2) on their domain. Then we obtain F as the union of all of these approximation functions. □

As in ZF this result can be generalised from the relation $<$ to arbitrary set-like well-founded relations.

Using GÖDEL-pairing one can also formalise ordered pairs of an ordinal α and a set z by

$$(\alpha, z) = \{G(\alpha, \beta) \mid \beta \in z\}.$$

One could now develop further mathematical notions - numbers, spaces, first-order syntax and semantics - in SO much the way as one does in standard set theory.

6 Assembling sets along wellfounded relations

In standard set theory a set x can be represented as a *point in a wellfounded relation*: consider the $\in$-relation on the transitive closure $\mathrm{TC}(\{x\})$ with distinguished element $x \in \mathrm{TC}(\{x\})$. By the MOSTOWSKI isomorphism theorem x is uniquely determined by the pair $(x, \mathrm{TC}(\{x\}))$ up to order isomorphism.

By the previous section, ordered pairs and wellfounded relations can be handled within the theory SO. *So assume* SO *for the following construction*. We shall eventually define a model of ZFC within SO.

Definition 33. *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.*

Note that according to our previous considerations one can reasonably define the class $\mathbb{P}$ in SO as well as in ZFC. In ZFC, $(x, \in \restriction \mathrm{TC}(\{x\}))$ is a point. Conversely, again in ZFC, any point $x = (x, R_x)$ can be interpreted as a standard set $I(x)$: 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

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

This means that the relation $I_x(u) = I_y(v)$ in the variables u and v can be defined recursively without actually forming the interpretations $I_x(u)$ and $I_x(v)$.

Hence this relation can be defined in SO.

Definition 34. *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 35. (SO) $\equiv$ *is an equivalence relation on $\mathbb{P}$.*

Proof. *Reflexivity*. Consider a point $x = (x, R_x)$. We show by induction on R_x that for all $u \in \mathrm{dom}(R_x)$ holds $(u, R_x) \equiv (u, R_x)$. Assume that the claim holds for all $v R_x u$. Consider some $v R_x u$. By the inductive assumption, $(v, R_x) \equiv (v, R_x)$. This implies

$$\forall v R_x u \exists w R_x u \, (v, R_x) \equiv (w, R_x).$$

By symmetry we also have

$$\forall w R_x u \, \exists v R_x u \ (v, R_x) \equiv (w, R_x).$$

Together these imply $(u, R_x) \equiv (u, R_x)$.

Symmetry. Consider points $x = (x, R_x)$ and $y = (y, R_y)$. We show by induction on the wellfounded relation $R_x \times R_y$ that

$$(u, R_x) \equiv (v, R_y) \ \text{ iff } \ (v, R_y) \equiv (u, R_x).$$

Assume that the claim holds for all (u', v') with $u' R_x u$ and $v' R_y v$. Assume that $(u, R_x) \equiv (v, R_y)$. To show that $(v, R_y) \equiv (u, R_x)$ consider $v' R_y v$. By assumption take $u' R_x u$ such that $(u', R_x) \equiv (v', R_y)$. By the inductive assumption on symmetry, $(v', R_y) \equiv (u', R_x)$. Hence

$$\forall v' R_y v \exists u' R_x u \ (v', R_y) \equiv (u', R_x).$$

Similarly

$$\forall u' R_x u \exists v' R_y v \ (v', R_y) \equiv (u', R_x)$$

and thus $(v, R_y) \equiv (u, R_x)$. This shows

$$(u, R_x) \equiv (v, R_y) \rightarrow (v, R_y) \equiv (u, R_x).$$

By the symmetry of the situation the implication from right to left also holds and

$$(u, R_x) \equiv (v, R_y) \leftrightarrow (v, R_y) \equiv (u, R_x).$$

In particular for $x = (x, R_x)$ and $y = (y, R_y)$

$$x \equiv y \leftrightarrow y \equiv x.$$

Transitivity. Consider points $x = (x, R_x)$, $y = (y, R_y)$ and $z = (z, R_z)$. We show by induction on the wellfounded relation $R_x \times R_y \times R_z$ that

$$(u, R_x) \equiv (v, R_y) \wedge (v, R_y) \equiv (w, R_z) \rightarrow (u, R_x) \equiv (w, R_z).$$

Assume that the claim holds for all (u', v', w') with $u' R_x u$, $v' R_y v$ and $w' R_z w$. Assume that

$$(u, R_x) \equiv (v, R_y) \wedge (v, R_y) \equiv (w, R_z).$$

To show that $(u, R_x) \equiv (w, R_z)$ consider $u' R_x u$. By $(u, R_x) \equiv (v, R_y)$ take $v' R_y v$ such that $(u', R_x) \equiv (v', R_y)$. By $(v, R_y) \equiv (w, R_z)$ take $w' R_z w$ such that $(v', R_y) \equiv (w', R_z)$. By the inductive assumption, $(u', R_x) \equiv (v', R_y)$ and $(v', R_y) \equiv (w', R_z)$ imply that $(u', R_x) \equiv (w', R_z)$. Thus

$$\forall u' R_x u \exists w' R_z w \ (u', R_x) \equiv (w', R_z).$$

Similarly

$$\forall w' R_z w \ \exists u' R_x u \ (u', R_x) \equiv (w', R_z)$$

and thus $(u, R_x) \equiv (w, R_z)$. In particular for $x = (x, R_x)$, $y = (y, R_y)$ and $z = (z, R_z)$

$$x \equiv y \wedge y \equiv z \rightarrow x \equiv z. \qquad \square$$

We now define a membership relation for points.

Definition 36. *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 37. (SO) *The equivalence relation $\equiv$ is a congruence relation with respect to $\blacktriangleleft$, i.e.,*

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

Proof. Let $x \blacktriangleleft y \wedge x \equiv x' \wedge y \equiv y' \ \rightarrow x' \blacktriangleleft y'$. Take $v R_y y$ such that $x \equiv (v, R_y)$. By $y \equiv y'$ take $v' R_{y'} y'$ such that $(v, R_y) \equiv (v', R_{y'})$. Since $\equiv$ is an equivalence relation, the equivalences $x \equiv x'$, $x \equiv (v, R_y)$ and $(v, R_y) \equiv (v', R_{y'})$ imply $x' \equiv (v', R_{y'})$. Hence $x' \blacktriangleleft y'$. $\square$

7 The class of points satisfies ZFC

We show that the class $\mathbb{P}$ of points with the relations $\equiv$ and $\blacktriangleleft$ satisfies the axioms ZFC of ZERMELO-FRAENKEL set theory with the axiom of choice. For the existence axioms of ZFC we prove a *comprehension lemma* about combining points into a single point.

Lemma 38. (SO) *Let* $(x_i\,|\,i\in A)$ *be a set-sized sequence of points. Then there is a point* $y=(y,R_y)$ *such that for all points* x *holds*

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

Proof. Obviously we may substitute the x_i's by $\equiv$-equivalent points x_i'. We may thus assume that the domains of the wellfounded relations R_{x_i} are pairwise disjoint. Take some $y\notin\bigcup_{i\in A}\mathrm{dom}(R_{x_i})$ and define the point $y=(y,R_y)$ by

$$R_y=\bigcup_{i\in A}R_{x_i}\cup\{(x_i,y)\,|\,i\in A\}.$$

Consider $i\in A$. If $x\in\mathrm{dom}(R_{x_i})$ then the iterated R_{x_i}-predecessors of x are equal to the iterated R_y-predecessors of x. Hence $(x,R_{x_i})\equiv(x,R_y)$.

Assume now that $x\blacktriangleleft y$. Take $v\,R_y\,y$ such that $x\equiv(v,R_y)$. Take $i\in A$ such that $v=x_i$. By the previous remark

$$x\equiv(v,R_y)=(x_i,R_y)\equiv(x_i,R_{x_i})=x_i\,.$$

Conversely consider $i\in A$ and $x\equiv x_i$. Then $x\equiv x_i=(x_i,R_{x_i})\equiv(x_i,R_y)$ and $x_i\,R_y\,y$. This implies $x\blacktriangleleft y$. □

We are now able to canonically interpret the theory ZFC within SO.

Theorem 39. (SO) $\mathbb{P}=(\mathbb{P},\equiv,\blacktriangleleft)$ *is a model of* ZFC.

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

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

Proof. Consider points x and y such that $\forall z(z\blacktriangleleft x\leftrightarrow z\blacktriangleleft y)$. Consider $u\,R_x\,x$. Then $(u,R_x)\blacktriangleleft(x,R_x)=x$. By assumption, $(u,R_x)\blacktriangleleft(y,R_y)$. By definition take $v\,R_y\,y$ such that $(u,R_x)\equiv(v,R_y)$. Thus

$$\forall u\,R_x\,x\,\exists v\,R_y\,y\,(u,R_x)\equiv(v,R_y).$$

By exchanging x and y one also gets

$$\forall v\,R_y\,y\,\exists u\,R_x\,x\,(u,R_x)\equiv(v,R_y).$$

Hence $x\equiv y$. *qed*(1)

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

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

Proof. Consider points $x=(x,R_x)$ and $y=(y,R_y)$. By the comprehension lemma 38 there is a point $z=(z,R_z)$ such that for all points w

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

qed(2)

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

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

Proof. Consider a point $x=(x,R_x)$. Let $A=\{i\in\mathrm{dom}(R_x)\,|\,\exists u\in\mathrm{dom}(R_x)\ i\,R_x\,u\,R_x\,x\}$. For $i\in A$ define the point $x_i=(i,R_x)$. By the comprehension lemma 38 there is a point $y=(y,R_y)$ such that for all points z

$$z\blacktriangleleft y\leftrightarrow\exists i\in A\,z\equiv x_i\,.$$

To show the axiom consider some $z\blacktriangleleft y$. Take $i\in A$ such that $z\equiv x_i$. Take $u\in\mathrm{dom}(R_x)$ such that $i\,R_x\,u\,R_x\,x$. Then $z\equiv x_i=(i,R_x)\blacktriangleleft(u,R_x)\blacktriangleleft(x,R_x)=x$, i.e., $\exists w(z\blacktriangleleft w\blacktriangleleft x)$.

Conversely assume that $\exists w(z\blacktriangleleft w\blacktriangleleft x)$ and take w such that $z\blacktriangleleft w\blacktriangleleft x$. Take $u\,R_x\,x$ such that $w\equiv(u,R_x)$. Then $z\blacktriangleleft(u,R_x)$. Take $i\,R_x\,u$ such that $z\equiv(i,R_x)=x_i$. Then $z\blacktriangleleft y$. *qed*(3)

(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 $\mathbb{P}$:

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

Proof. Note that the formula φ may contain further free parameters, which we do not mention for the sake of simplicity. Assume that $\forall u, v, v' ((\varphi(u, v) \wedge \varphi(u, v')) \rightarrow v \equiv v')$ and let $x = (x, R_x)$ be a point. Let $A = \{i \mid i R_x\, x\}$. For each $i \in A$ we have the point $(i, R_x) \blacktriangleleft (x, R_x) = x$. Using replacement and choice in SO we can pick for each $i \in A$ a point $z_i = (z_i, R_{z_i})$ such that $\varphi((i, R_x), z_i)$ holds if such a point exists. By the comprehension lemma 38 there is a point $y = (y, R_y)$ such that for all points z

$$z \blacktriangleleft y \leftrightarrow \exists i \in A\, z \equiv z_i .$$

To show the instance of the replacement schema under consideration, assume that $z \blacktriangleleft y$. Take $i \in A$ such that $z \equiv z_i$. Then $(i, R_x) \blacktriangleleft (x, R_x) = x$, $\varphi((i, R_x), z_i)$ and $\varphi((i, R_x), z)$. Hence $\exists u (u \blacktriangleleft x \wedge \varphi(u, z))$.

Conversely, assume that $\exists u (u \blacktriangleleft x \wedge \varphi(u, z))$. Take $u \blacktriangleleft x$ such that $\varphi(u, z)$. Take $i R_x\, x$, $i \in A$ such that $u \equiv (i, R_x)$. Then $\varphi((i, R_x), z)$. By definition of z_i , $\varphi((i, R_x), z_i)$. The functionality of the formula φ implies $z \equiv z_i$. Hence $\exists i \in A\, z \equiv z_i$ and $z \blacktriangleleft y$. *qed*(4)

The replacement schema also implies the *separation* schema.

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

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

Proof. By the separation schema it suffices to show that

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

Consider a point $x = (x, R_x)$. Let $F = \mathrm{dom}(R_x) \cup \mathrm{ran}(R_x)$ be the *field* of R_x . By the powerset axiom of SO choose some set P such that $\mathrm{Pow}(P, F)$:

$$\forall z (\exists \alpha (\alpha \in z) \wedge \forall \alpha (\alpha \in z \rightarrow \alpha \in F) \rightarrow \exists \xi \forall \beta (\beta \in z \leftrightarrow (\beta, \xi) \in P)).$$

Choose two large ordinals δ and y such that

$$\forall \alpha \in F\, \alpha < \delta \text{ and } \forall \xi\, (\xi \in \mathrm{ran}(P) \rightarrow (\delta, \xi) < y).$$

Define a point $y = (y, R_y)$ by

$$R_y = R_x \cup \{(\beta, (\delta, \xi)) \mid (\beta, \xi) \in P\} \cup \{((\delta, \xi), y) \mid \xi \in \mathrm{ran}(P)\}.$$

To show the axiom consider some point $c = (c, R_c)$ such that $\forall w (w \blacktriangleleft c \rightarrow w \blacktriangleleft x)$. Define a corresponding subset z of F by

$$z = \{\beta \in F \mid \exists v R_c\, c\, (v, R_c) \equiv (\beta, R_x)\}.$$

We may assume for simplicity that $z \neq \emptyset$. By the powerset axiom of SO choose $\xi \in \mathrm{ran}(P)$ such that

$$\forall \beta (\beta \in z \leftrightarrow (\beta, \xi) \in P).$$

We claim that $((\delta, \xi), R_y) \equiv c$ and thus $c \blacktriangleleft y$.

Consider $\beta R_y\, (\delta, \xi)$. By the definition of R_y we have $(\beta, \xi) \in P$ and so $\beta \in z$. By the definition of z choose $v R_c\, c$ such that $(v, R_c) \equiv (\beta, R_x) \equiv (\beta, R_y)$.

Conversely, consider $v R_c\, c$. Then $(v, R_c) \blacktriangleleft (c, R_c) = c$. The subset property implies $(v, R_c) \blacktriangleleft (x, R_x) = x$. Take $\beta R_x\, x$ such that $(v, R_c) \equiv (\beta, R_x) \equiv (\beta, R_y)$. By definition, $\beta \in z$, $(\beta, \xi) \in P$ and $\beta R_y\, (\delta, x)$. *qed*(5)

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

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

Proof. Let $x = (x, R_x) \in \mathbb{P}$ be a point such that

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

Choose an ordinal $\alpha \in \mathrm{dom}(R_x)$ and define the point $w = (\alpha, R_w)$ by letting its "elements" be least ordinals in the "elements" of x:

$$R_w = R_x \cup \{(\xi, \alpha) \mid \exists \zeta (\xi R_x \zeta R_x x \wedge (\forall \xi' < \xi \ \forall \zeta' ((\zeta, R_x) \equiv (\zeta', R_x) \rightarrow \neg(\xi R_x \xi' R_x \zeta))))\}.$$

To show that w witnesses the axiom of choice for x consider a point y with $y \blacktriangleleft x$. We may assume that y is of the form $y = (\zeta, R_x)$ where $\zeta R_x x$. By the assumption on x there exists $u \blacktriangleleft y$. Take some ξ such that $(\xi, R_x) \equiv u$. We may assume that ζ and ξ with these properties are chosen so that ξ is minimal in the ordinals. Then

$$\xi R_x \zeta R_x x \wedge (\forall \xi' < \xi \ \forall \zeta' ((\zeta, R_x) \equiv (\zeta', R_x) \rightarrow \neg(\xi R_x \xi' R_x \zeta))) \tag{3}$$

and so $\xi R_w \alpha$. Thus $u \blacktriangleleft w$. To show the uniqueness of this u with $u \blacktriangleleft w \wedge u \blacktriangleleft y$ consider some v with $v \blacktriangleleft w \wedge v \blacktriangleleft y$. We may assume that v is of the form $v = (\xi', R_w)$ with $\xi' R_w \alpha$. By the definition of R_w we choose some ζ' such that

$$\xi' R_x \zeta' R_x x \wedge (\forall \xi'' < \xi' \ \forall \zeta'' ((\zeta', R_x) \equiv (\zeta'', R_x) \rightarrow \neg(\xi' R_x \xi'' R_x \zeta'))). \tag{4}$$

Now

$$v \blacktriangleleft y \blacktriangleleft x \text{ and } v = (\xi', R_w) \blacktriangleleft (\zeta', R_w) \blacktriangleleft (x, R_x) = x.$$

Since the "elements" of x are "pairwise disjoint", we have $y \equiv (\zeta', R_w)$. Since $y \equiv (\zeta, R_x)$ the conditions (2) and (3) become equivalent and define the same ordinal $\xi = \xi'$. Hence

$$u \equiv (\xi, R_x) \equiv (\xi', R_w) \equiv v.$$

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 $\mathbb{P}$:

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

Proof. Note that the formula φ may contain further free parameters, which we do not mention for the sake of simplicity. Assume that $\exists u \varphi(u)$. Take a point $x = (x, R_x)$ such that $\varphi(x)$. Since R_x is wellfounded one may take an R_x-minimal $y \in \mathrm{dom}(R_x)$ such that $\varphi((y, R_x))$. Letting y also denote the point (y, R_x) then $\varphi(y)$. To prove the axiom, consider some point $z \blacktriangleleft y$. Take $v R_x y$ such that $z \equiv (v, R_x)$. By the R_x-minimal choice of y we have $\neg \varphi((v, R_x))$. Hence $\neg \varphi(z)$. *qed*(7)

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

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

Proof. In SO let ω be the smallest limit ordinal. We show that

$$x = (\omega, < \restriction (\omega + 1)^2)$$

witnesses the axiom. Since $(0, < \restriction (\omega + 1)^2) \blacktriangleleft (\omega, < \restriction (\omega + 1)^2)$ we have $\exists y\, y \blacktriangleleft x$. Consider some $y \blacktriangleleft x$. We may assume that $y = (n, < \restriction (\omega + 1)^2)$ for some $n < \omega$. Set

$$z = (n + 1, < \restriction (\omega + 1)^2).$$

It is easy to check that

$$z \blacktriangleleft x \wedge \forall u (u \blacktriangleleft z \leftrightarrow u \blacktriangleleft y \vee u \equiv y).$$

qed(8) □

Theorem 40. *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 $\mathcal{S} = \{v \in M \mid v \subseteq \mathrm{Ord}\}$.*

Proof. Define the model $\mathbb{P}=(\mathbb{P},\equiv,\blacktriangleleft)$ from $(\mathrm{Ord},\mathcal{S},<,=,\in,G)$ as above. Consider a point $x=(x,R_x)\in\mathbb{P}$. Then x is also an ordinal in the sense of V. In $\mathcal{S}$, apply the recursion theorem to the wellfounded relation R_x and obtain an order-preserving map

$$\sigma\colon(\mathrm{dom}(R_x),R_x)\to(\mathrm{Ord},<).$$

Transfer the map σ to V by defining

$$\tilde{\sigma}=\{(\alpha,\beta)\,|\,\mathcal{S}\vDash\sigma(\alpha)=\beta\}\colon\mathrm{dom}(R_x)\to\mathrm{Ord}\,.$$

This map is order-preserving and witnesses that R_x is wellfounded in V. So (x,R_x) is a point in the sense of V. In V, define the interpretation function $I\colon\mathbb{P}\to V$ recursively by

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

Set

$$M=\{I(x)\,|\,x\in\mathbb{P}\}.$$

(1) M is transitive.
Proof. Consider $y\in I(x)\in M$. Choose $v\,R_x\,x$ such that $y=I_x(v)$. Then $(v,R_x)\in\mathbb{P}$ and

$$y=I_x(v)=I((v,R_x))\in M\,.$$

qed (1)

The above definitions imply:

(2) The function $I\colon\mathbb{P}\to M$ is surjective and preserves $\equiv$ and $=$, and $\blacktriangleleft$ and $\in$, resp.:

$$\forall x,y\in\mathbb{P}\colon((x\equiv y\leftrightarrow I(x)=I(y))\wedge(x\blacktriangleleft y\leftrightarrow I(x)\in I(y))).$$

Hence

(3) M is a transitive $\in$-model of the ZFC-axioms, i.e., M is an inner model.
(4) $\mathcal{S}=\{v\in M\,|\,v\subseteq\mathrm{Ord}\}$.
Proof. Let $v\in\mathcal{S}$. We build a point that will be interpreted as v. Choose an ordinal α such that $v\subseteq\alpha$. Define a wellfounded relation R_x on $\alpha+1$ by

$$\xi\,R_x\,\zeta\text{ iff }(\xi<\zeta<\alpha\text{ or }(\zeta=\alpha\wedge\xi\in v)).$$

Then $x=(\alpha,R_x)$ is a point. Let $I_x(u)=\{I_x(v)\,|\,v\,R_x\,u\}$ be the recursive interpretation function for x. For $\zeta<\alpha$ we have $I_x(\zeta)=\zeta$ since we have inductively

$$I_x(\zeta)=\{I_x(\xi)\,|\,\xi\,R_x\,\zeta\}=\{\xi\,|\,\xi<\zeta\}=\zeta\,.$$

And then

$$I(x)=I_x(\alpha)=\{I_x(\xi)\,|\,\xi\,R_x\,\alpha\}=\{\xi\,|\,\xi\in v\}=v\,.$$

Hence $v=I(x)\in M$.

The previous argument also shows that one may canonically represent an ordinal ξ by the point $(\xi,<\restriction(\xi+1)^2)$:

$$I((\xi,<\restriction(\xi+1)^2))=\xi\,.$$

For the converse inclusion consider some $v\in M$, $v\subseteq\alpha\in\mathrm{Ord}$. Choose a point $x\in\mathbb{P}$ such that $I(x)=v$. Since $\mathcal{S}$ satisfies the separation schema,

$$v=\{\xi<\alpha\,|\,\xi\in v\}=\{\xi<\alpha\,|\,\mathcal{S}\vDash(\xi,<\restriction(\xi+1)^2)\blacktriangleleft x\}\in\mathcal{S}.$$

qed(4)

The model M is unique since it is determined by its sets of ordinals (see [6], Theorem 13.28). □

8 An ordinal computable truth predicate

We shall later define a truth predicate whose recursive definition is of the form

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

We show that such recursions can be carried out within the collection of ordinal register computable functions: if H is computable then the recursive function F is computable. The computation will be based on a *stack* which can hold finite decreasing sequences of ordinals and some other information.

8.1 3-adic representations and ordinal stacks

We intend to use a *stack* that can hold a (finite) sequence $\alpha_0 > \alpha_1 > \ldots > \alpha_{n-2} \geqslant \alpha_{n-1}$ of ordinals which is strictly decreasing except possibly for the last two ordinals. This sequence of ordinals can coded by the ordinal $\alpha = 3^{\alpha_0} + 3^{\alpha_1} + \ldots + 3^{\alpha_{n-1}}$. We state some facts of ordinal arithmetic and define computable functions for dealing with ordinals as stacks.

Proposition 41. *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.*

Proof. Assume the property for $\beta < \alpha$. Since the ordinal exponentiation $\nu \mapsto \delta^{\nu}$ is continuous and strictly monotone there is a maximal $\alpha_0 \leqslant \alpha$ such that $\delta^{\alpha_0} \leqslant \alpha$. Then

$$\delta^{\alpha_0+1} = \delta^{\alpha_0} \cdot \delta > \alpha.$$

Since the ordinal multiplication $\nu \mapsto \delta^{\alpha_0} \cdot \nu$ is continuous and strictly monotone there is a largest ζ_0, $0 < \zeta_0 < \delta$ such that $\delta^{\alpha_0} \cdot \zeta_0 \leqslant \alpha$. Then

$$\delta^{\alpha_0} \cdot (\zeta_0 + 1) = \delta^{\alpha_0} \cdot \zeta_0 + \delta^{\alpha_0} > \alpha.$$

Since the ordinal addition $\nu \mapsto \delta^{\alpha_0} \cdot \zeta_0 + \nu$ is an increasing enumeration of the ordinals $\geqslant \delta^{\alpha_0} \cdot \zeta_0$ there is $\beta < \delta^{\alpha_0} \leqslant \alpha$ such that $\alpha = \delta^{\alpha_0} \cdot \zeta_0 + \beta$. By the inductive assumption, β has a δ-adic representation $\beta = \delta^{\alpha_1} \cdot \zeta_1 + \ldots + \delta^{\alpha_{n-1}} \cdot \zeta_{n-1}$. Since $\beta < \delta^{\alpha_0}$ we have $\alpha_1 < \alpha_0$. Thus

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

is a δ-adic representation of α.

We show uniqueness. Assume that also

$$\alpha = \delta^{\alpha'_0} \cdot \zeta'_0 + \delta^{\alpha'_1} \cdot \zeta'_1 + \ldots + \delta^{\alpha'_{r-1}} \cdot \zeta'_{r-1}$$

Then $\alpha \in [\delta^{\alpha_0} \cdot \zeta_0, \delta^{\alpha_0} \cdot (\zeta_0 + 1))$ and $\alpha \in [\delta^{\alpha'_0} \cdot \zeta'_0, \delta^{\alpha'_0} \cdot (\zeta'_0 + 1))$. By the pairwise disjointness of intervals of this type, $\alpha_0 = \alpha'_0$ and $\zeta_0 = \zeta'_0$. Furthermore

$$\delta^{\alpha_1} \cdot \zeta_1 + \ldots + \delta^{\alpha_{n-1}} \cdot \zeta_{n-1} = \delta^{\alpha'_1} \cdot \zeta'_1 + \ldots + \delta^{\alpha'_{r-1}} \cdot \zeta'_{r-1} < \delta^{\alpha_0} \cdot \zeta_0 \leqslant \alpha.$$

By the inductive assumption, $(\alpha_1, \ldots, \alpha_{n-1}) = (\alpha'_1, \ldots, \alpha'_{r-1})$ and $(\zeta_1, \ldots, \zeta_{n-1}) = (\zeta'_1, \ldots, \zeta'_{n-1})$. Hence the two δ-adic representation of α agree. □

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

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

We call the natural number n the *length* of the stack α . The final elements $\alpha_{n-1}, \alpha_{n-2}, \ldots$ of this stack can be defined from α as follows:

$$\begin{aligned} \alpha_{n-1} &= \text{the largest } \xi \text{ such that there is } \zeta \text{ with } \alpha = 3^{\xi} \cdot \zeta \\ \alpha_{n-2} &= \text{the largest } \xi \text{ such that there is } \zeta \text{ with } \alpha - 3^{\alpha_{n-1}} = 3^{\xi} \cdot \zeta \\ &\ldots \end{aligned}$$

The ordinals α_{n-1}, α_{n-2} are obviously ordinal register computable by some programs `last`, `llast` resp., we agree that these functions return a special value `UNDEFINED` if the stack is too short.

If the stack α is kept in a register `stack` then the liminf behaviour of registers implies the following crucial limit behaviour of stacks.

Proposition 42. *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 $(\rho(\tau) | \tau \in [t_0, t))$ 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 $\rho = \bigcup_{\tau \in [t_0, t)} \rho(\tau)$.*

8.2 Stack recursion

Given an algorithm for the recursion function H we compute F with a `stack` as considered above and a variable `value` which can hold a single value of the function F: we let `value = 2` stand for "undefined'. The *intention* of the following program P is to accept an input ordinal α on the singleton stack α and stop with the output stack α and `value` $= F(\alpha)$. During the recursion the program will call itself with non-empty stacks $\alpha = \alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ and compute the value $F(\alpha_{n-1})$. During the main loop of the program the bounded quantifier $\exists \nu < \alpha$ ranges over all $\nu < \alpha$. The subloop evaluates the kernel $H(\alpha, \nu, F(\nu)) = 1$ of the quantifier and returns the result for the further calculation.

```
  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
```

The correctness of the program with respect to the above intention is established by

Theorem 43. *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.*

Proof. a) and b) are proved simultaneously by induction over the last element α_{n-1} of the stack. Assume that P is in state `MainLoop` at time s with stack contents $\langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}\rangle$ where $n \geqslant 1$ and that a) and b) hold for all stack contents $\langle\beta_0, \beta_1, \ldots, \beta_{m-1}\rangle$ with $\beta_{m-1} < \alpha_{n-1}$. Define $\bar{\alpha}$ as in b).

We first prove b) by defining an appropriate sequence $(t_i | i \leqslant \bar{\alpha})$ by recursion over $i \leqslant \bar{\alpha}$.
$i = 0$: inspection of P shows that the computation will move to state `2` and obtain `stack` contents $\langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, 0\rangle$ before immediately returning to `MainLoop`.
$i = j + 1$ where $j < \bar{\alpha}$. By recursion, P is in state `MainLoop` at time t_j with stack contents $\langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, j\rangle$. $j < \bar{\alpha} \leqslant \alpha_{n-1}$ so that by the simultaneous induction a) holds for $\langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, j\rangle$. So there will be a later time when P is in state `SubLoop` with stack contents $\langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, j\rangle$ and `value` $= F(j)$. Also during that computation the stack contents will always be $\geqslant \langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, j\rangle$. Inspection of the program shows that it will further compute $H(\alpha_{n-1}, j, F(j))$. This value will be $\neq 1$ by definition of $\bar{\alpha}$. So the computation will move on to state `4` with stack contents $\langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, j+1\rangle$. At the subsequent time $t_i = t_{j+1}$ P is in state `MainLoop` with stack contents $\langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, i\rangle$.
i is a limit ordinal. Then by the limit behaviour of the machine and in particular by the above proposition, at time $\bigcup \{t_j | j < i\}$ the machine will be in state `MainLoop` with `stack` contents.

Now we prove a).
Case 1: $\bar{\alpha} < \alpha_{n-1}$. Then $F(\bar{\alpha}) = 1$. By b) P will get to state `MainLoop` with `stack` contents $\langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, \bar{\alpha}\rangle$. By the inductive hypothesis, P will then get to state `SubLoop` with `stack` contents $\langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}\rangle$ and `value` set to $F(\bar{\alpha})$. Then the program will compute $H(\alpha_{n-1}, \bar{\alpha}, F(\bar{\alpha})) = 1$ and move into alternative `3`. The register `value` obtains the value $F(\alpha_{n-1}) = 1$ and the program moves to state `SubLoop` with the last stack element removed: `stack` $= \langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}\rangle$, as required.
Case 2: $\bar{\alpha} = \alpha$. Then $F(\bar{\alpha}) = 0$. By b), P will get to state `MainLoop` with stack contents $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, \bar{\alpha} = \alpha_{n-1}$. Inspection of the program shows that it will get into alternative `1`, set `stack:` $= \langle\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}\rangle$, `value` $= 0$ and move to `SubLoop`, which proves a) in this case.

Finally, c) follows readily from a) and inspection of the program. □

8.3 A recursive truth predicate

The GÖDEL pairing function G allows to code finite sequences $\alpha_0, \ldots, \alpha_{n-1}$ of ordinals into single ordinals. The coding can be made computable in the sense that usual operations on finite sequences like concatenation or substitution are computable as well. By these techniques one can also code formal languages in a computable way.

We shall consider a language L_R appropriate for first-order structures of the form

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

where the GÖDEL function G is viewed as a ternary relation on α and R is a unary relation on α. The terms of the language are variables v_n for $n<\omega$ and constant symbols c_ξ for $\xi\in\mathrm{Ord}$; the symbol c_ξ will be interpreted as the ordinal ξ. 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; thus we are only working with bounded quantification. We arrange the computable coding in a way 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 $\zeta<\xi$:

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

An L_R-formula is an L_R-*sentence* if it does not have free variables. If φ is an L_T-sentence so that all constants symbols c_ξ in φ have indices $\xi<\alpha$ then the satisfaction relation

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

is defined as usual. Bounded sentences are *absolute* for sufficiently long initial segments of the ordinals. If φ is a bounded sentence such that every constant symbol c_ξ occuring in φ satisfies $\xi<\beta<\alpha$ then

$$(\alpha,<,G,R)\vDash\varphi\text{ iff }(\beta,<,G,R)\vDash\varphi.$$

The coding of formulas by ordinals φ will satisfy that $\xi<\varphi$ for every constant symbol c_ξ occuring in φ. So the meaning of a bounded sentence φ is given by

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

This leads to the recursive definition of a *bounded truth predicate* $T\subseteq\mathrm{Ord}$ over the ordinals

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

We shall later see that T is a strong predicate which codes a model of set theory. We show that the characteristic function χ_T of T can be defined according to the recursion scheme

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

of section 5 and is thus ordinal register computable provided we can exhibit an appropriate computable recursion function H:

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

Assuming that the syntactical operations are computable, H and thus the bounded truth predicate T are computable.

9 Computing a model of set theory

The truth predicate T contains information about a large class of sets of ordinals.

Definition 44. *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 45. $(\mathrm{Ord},\mathcal{S},<,=,\in,G)$ *is a model of the theory* SO.

Proof. The axioms (1)-(7) are obvious. The proofs of axiom schemas (8) and (9) rest on a LEVY-type reflection principle. For $\theta\in\mathrm{Ord}$ define

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

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

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

Since all elements of $\mathcal{S}_\theta$ can be defined from the truth function T and ordinals $<\theta$, the right-hand side can be evaluated in the structure $(\theta,<,G\cap\theta^3,T)$ by an L_T-formula φ^* which can be recursively computed from φ. Hence

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

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

The powerset axiom can be shown by a similar reflection argument. □

9.1 Ordinal computability corresponds to constructibility

KURT GÖDEL [4] defined the inner model L of *constructible sets* as the union of a hierarchy of levels L_α:

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

where the hierarchy is defined by: $L_0=\emptyset$, $L_\delta=\bigcup_{\alpha<\delta}L_\alpha$ 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. The standard reference for the theory of the model L is the monograph [3].

The following main result provides a characterization of ordinal register computability which does not depend on a specific machine model or coding of language:

Theorem 46. *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 $\delta_1,\dots,\delta_{n-1}$, so that for every $\alpha\in\mathrm{Ord}$:

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

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

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

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

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

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. □

10 An application: the generalised continuum hypothesis in L

Ordinal computability allows to reprove some basic facts about the constructible universe L. The analogue of the axiom of constructibility, $V = L$, is the statement that *every* set of of ordinals is ordinal computable.

Theorem 47. *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. □

The following theorem is proved by a condensation argument for ordinal computations which is a simple analogue of the usual condensation argument for the constructible hierarchy.

Theorem 48. *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, $\mathrm{card}(\bar{M}) = \kappa$ 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 $\mathrm{card}(\bar{M}) < \kappa^+$. 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 $\alpha_1, \ldots, \alpha_{n-1} < \kappa^+$.
b) follows from a) since there are a countable many programs and κ^+ many finite sets of ordinals $< \kappa^+$.
c) is immediate from b). □

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.