Ordinal Machines and Combinatorial Principles

by Peter Koepke

University of Bonn

Wien, November 19, 2005

- Register machines working with ordinals
- computable = constructible
- the Continuum Hypothesis in L
- Silver machines
- defining a Silver machine from a register machine
- ...

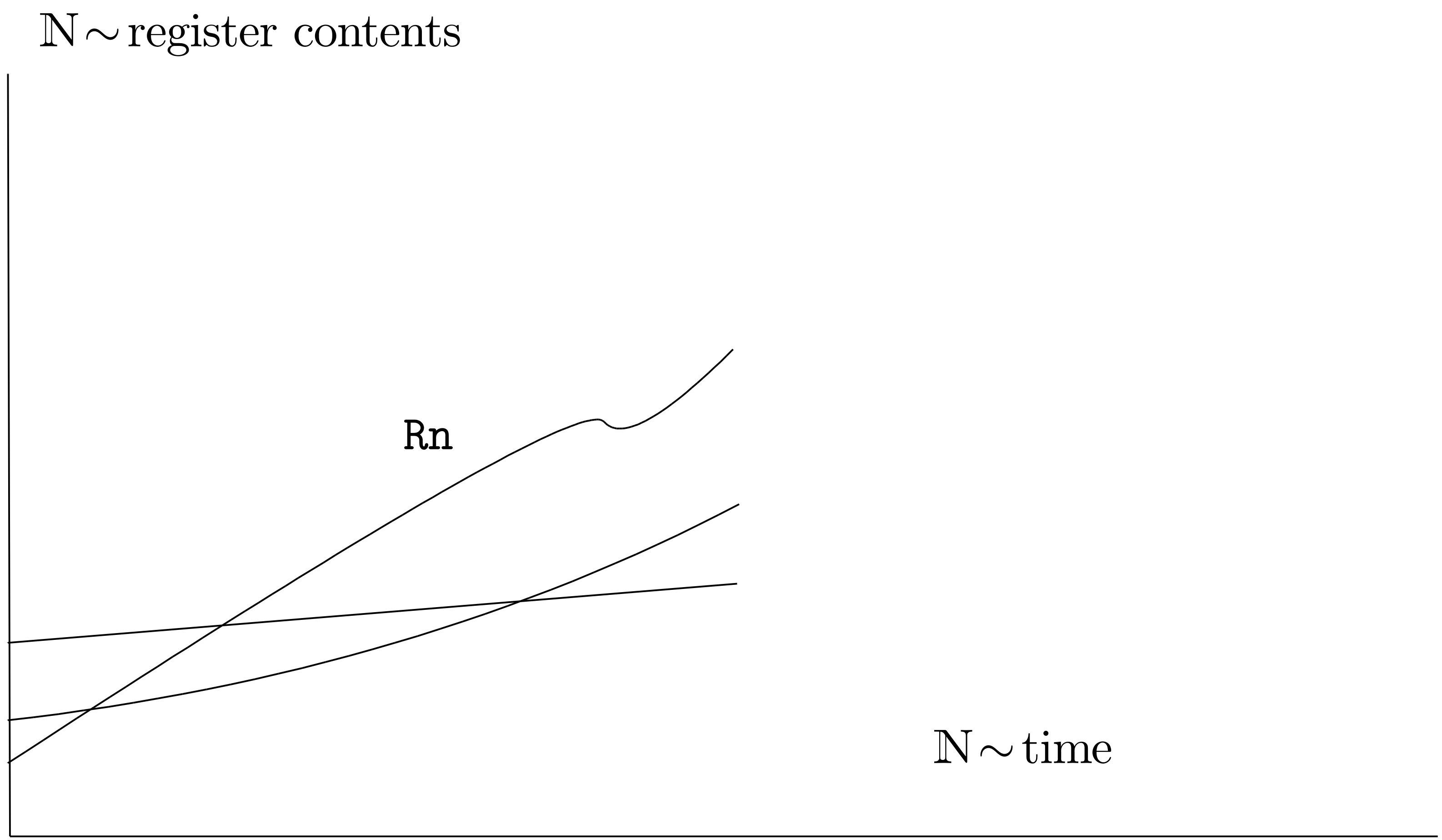
A register program for multiplication

```
Addition, computing R2 = R0 + R1:
0  R3:=0
1  R4:=0
2  R2:=0
3  if R0=R3 then go to 7
4  R3:=R3+1
5  R2:=R2+1
6  go to 3
7  if R1=R4 then STOP
8  R4:=R4+1
9  R2:=R2+1
10 go to 7
```

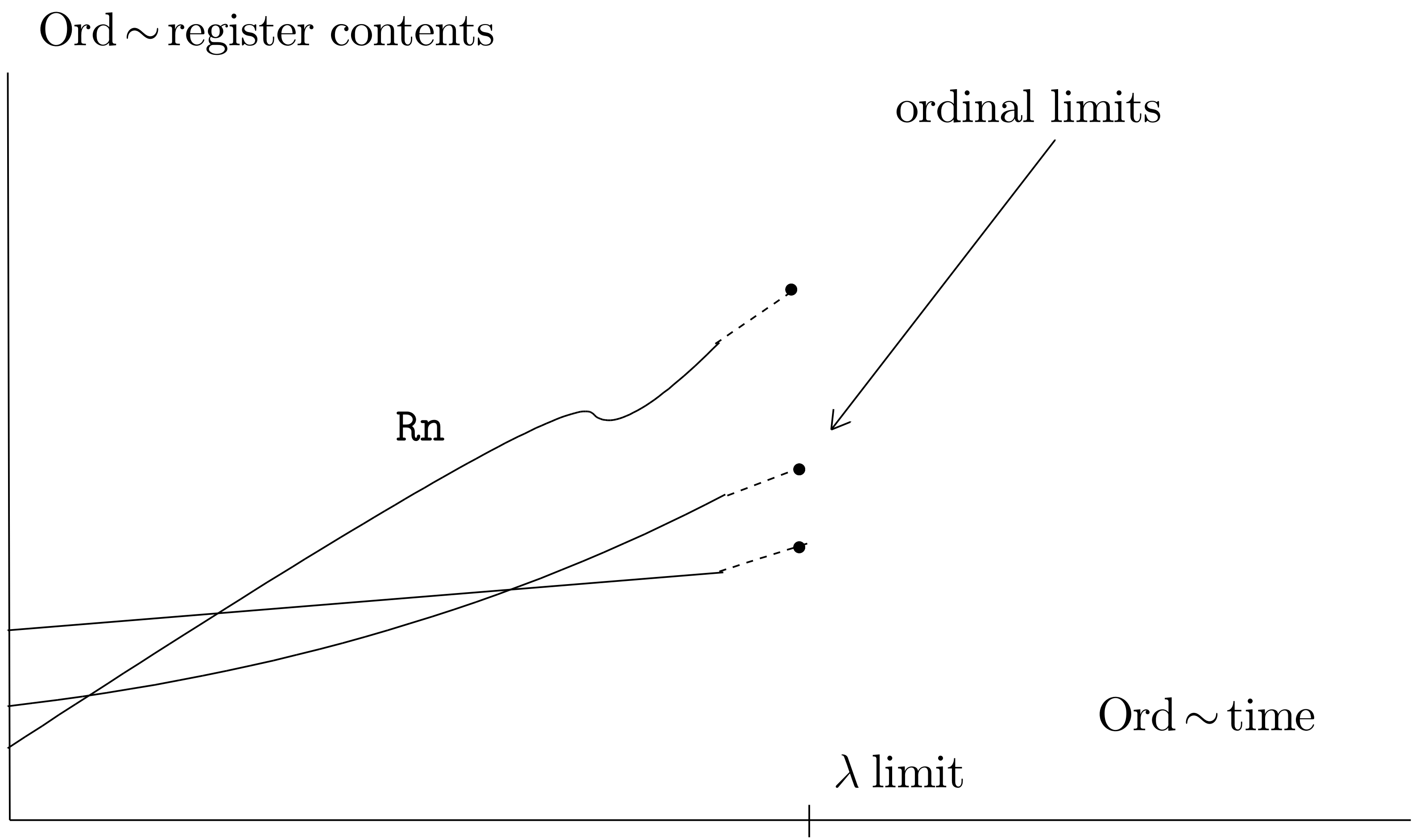
Or, with names for registers

```
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
```

A picture of a computation:



Computing with ordinals (Ryan Siders, Helsinki):



Then, the previous program performs the standard ordinal addition:

```
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
```

Ordinal addition is *(ordinal) computable*.

Let P be a register program and let $\vec{\alpha} = (\alpha_0, \alpha_1, ..., 0, 0, ...)$ and $\vec{\beta} = (\beta_0, \beta_1, ..., 0, 0, ...)$ be ω-sequences of ordinals which eventually vanish. Then

$$P(\vec{\alpha}) = \vec{\beta}$$

expresses that the program P when started with the initial register contents $\vec{\alpha}$ stops and then the register contents are $\vec{\beta}$. If the program does not stop we write

$$P(\vec{\alpha}) = \uparrow$$

A function F: Ord $\rightarrow 2$ is *(ordinal) computable* by the program P and the parameters $\beta_1, ..., \beta_n$ if

$$\forall \beta_0 \, P(\beta_0, \beta_1, ..., \beta_n, 0, 0, ...) = (F(\beta_0), ...)$$

A set $x \subseteq$ Ord is *(ordinal) computable* if its characteristic function is *ordinal computable* (by some program and ordinal parameters).

The standard Gödel pairing for ordinals is computable:

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

GÖDEL's constructible hierarchy

- $L_0 = \emptyset$
- $L_{\alpha+1} = \{\{x \in L_\alpha \mid L_\alpha \vDash \varphi(x, \vec{y})\} \mid \varphi \in \mathrm{Fml}, \vec{y} \in L_\alpha\}$
- $L_\lambda = \bigcup_{\alpha<\lambda} L_\alpha$
- $L = \bigcup_{\alpha \in \mathrm{Ord}} L_\alpha$

Everything in L is *named* by finitely many ordinals:

$$\begin{aligned}\{x \in L_\alpha \mid L_\alpha \vDash \varphi(x, \vec{y})\} &= \{x \in L_\alpha \mid L_\alpha \vDash \varphi(x, \{x \in L_\beta \mid L_\beta \vDash \psi(x, \vec{z})\})\} \\ &= \{x \in L_\alpha \mid L_\alpha \vDash \varphi(x, \{x \in L_\beta \mid L_\beta \vDash \psi(x, \{\ldots\})\})\} \\ &\sim (\alpha, \varphi, \beta, \psi, \ldots)\end{aligned}$$

This corresponds to *one* ordinal via GÖDEL pairing.

Computing bounded truth in L

$$L_{\alpha+1} \models x \in \{x \in L_\alpha \mid L_\alpha \models \varphi(x, \vec{y})\}$$
$$\leftrightarrow \; L_\alpha \models \varphi(x, \vec{y})$$
$$\leftrightarrow \; L_\alpha \models (\psi_0 \vee \psi_1)(x, \vec{y})$$
$$\leftrightarrow \; L_\alpha \models \psi_0(x, \vec{y}) \text{ or } L_\alpha \models \psi_1(x, \vec{y})$$
$$\leftrightarrow \; \ldots$$

$$L_\alpha \models \exists v \psi(v, \vec{y})$$
$$\leftrightarrow \; \text{there is } x \in L_\alpha \text{ such that } L_\alpha \models \psi(x, \vec{y})$$
$$\leftrightarrow \; \ldots$$

One can arrange, that the RHS formulas are smaller in an adequate well-order.

If we define a *constructible truth predicate* $F\colon \mathrm{Ord} \to 2$ by

$$F(\ulcorner L_\alpha \models \varphi(x, \vec{y}\,)\urcorner) = 1 \text{ iff } L_\alpha \models \varphi(x, \vec{y}\,)$$

then F has a recursive definition of the form:

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

for some computable function H.

Computing $F(3)$ with a stack of ordinals:

time	stack contents	numerical code
1	$F(3)?$	$2^3 = 8$
2	$F(3)?, F(0)?$	$2^3 + 2^0 = 9$
3	$F(3)?, F(0)!(= 0)$	
4	$F(3)?, F(1)?$	$2^3 + 2^1 = 10$
5	$F(3)?, F(1)?, F(0)?$	$2^3 + 2^1 + 2^0 = 11$
6	$F(3)?, F(1)?, F(0)!$	
7	$F(3)?, F(1)!$	
8	$F(3)!$	

Computing $F(\omega+2)$ with a stack of ordinals:

time	stack contents	ordinal code
1	$F(\omega+2)?$	$2^{\omega+2}$
2	$F(\omega+2)?, F(0)?$	$2^{\omega+2}+2^0$
3	$F(\omega+2)?, F(0)!(=0)$	
4	$F(\omega+2)?, F(1)?$	$2^{\omega+2}+2^1$
5	$F(\omega+2)?, F(1)?, F(0)?$	$2^{\omega+2}+2^1+2^0$
6	$F(\omega+2)?, F(1)?, F(0)!$	
7	$F(\omega+2)?, F(1)!$	
$\vdots$	$\vdots$	$\vdots$
	$F(\omega+2)?, F(n)?$	$2^{\omega+2}+2^n$
	$\vdots$	
	$F(\omega+2)?, F(n)!$	
	$\vdots$	
	$F(\omega+2)?, F(\omega)?$	$2^{\omega+2}+2^\omega = \lim_{n<\omega}(2^{\omega+2}+2^n)$
	$\vdots$	

Hence the constructible truth predicate is ordinal computable.

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

Proof. $(\rightarrow)$. Let

$$\forall \beta_0 \, P(\beta_0, \beta_1, \ldots, \beta_n, 0, 0, \ldots) = (\chi_x(\beta_0), \ldots).$$

By the absoluteness of ordinal computations,

$$\forall \beta_0 \, (P(\beta_0, \beta_1, \ldots, \beta_n, 0, 0, \ldots))^L = (\chi_x(\beta_0), \ldots).$$

Hence x is definable in L.

$(\leftarrow)$. Let $x = \{v \in L_\alpha \mid L_\alpha \vDash \varphi(v, \vec{\beta}\,)\}$. Then χ_x is computable:

$$\begin{aligned}\chi_x(\xi) = 1 \quad &\text{iff} \quad L_\alpha \vDash \varphi(\xi, \vec{\beta}\,) \\ &\text{iff} \quad F\left(\left\ulcorner L_\alpha \vDash \varphi(\xi, \vec{\beta}\,) \right\urcorner\right) = 1 \\ &\text{iff} \quad F(G(\xi, \alpha, \vec{\beta}\,)) = 1\end{aligned}$$

where G is the computable function $(\xi, \alpha, \vec{\beta}\,) \mapsto \left\ulcorner L_\alpha \vDash \varphi(\xi, \vec{\beta}\,) \right\urcorner$. $\square$

The *axiom of constructibility* $V = L$ from the viewpoint of ordinal computability:

$$\forall x \subseteq \mathrm{Ord}\, \exists P\, \exists \beta_1, \ldots, \beta_n\, \forall \beta_0\, (\beta_0 \in x \leftrightarrow P(\beta_0, \beta_1, \ldots, \beta_n, 0, 0, \ldots) = 1).$$

Theorem 2. $V = L \rightarrow \mathrm{CH}$.

Proof. Let $x \subseteq \omega$. By $V = L$ let

$$\forall \beta_0\, (\beta_0 \in x \leftrightarrow P(\beta_0, \beta_1, \ldots, \beta_n, 0, 0, \ldots) = 1).$$

Let $X \prec V$ be countable, $x, \beta_1, \ldots, \beta_n \in X$. Let

$$\pi \colon (X, \in) \cong (M, \in),$$

M transitive, $\bar{\beta}_1 = \pi(\beta_1), \ldots, \bar{\beta}_n = \pi(\beta_n)$. Then, by absoluteness,

$$\forall \beta_0\, (\beta_0 \in x \leftrightarrow (P(\beta_0, \bar{\beta}_1, \ldots, \bar{\beta}_n, 0, 0, \ldots) = 1)^M \leftrightarrow P(\beta_0, \bar{\beta}_1, \ldots, \bar{\beta}_n, 0, 0, \ldots) = 1).$$

Hence x is defined by the program P and the parameters $\bar{\beta}_1, \ldots, \bar{\beta}_n < \aleph_1$. There are $\aleph_1$ such definitions, hence $\mathrm{card}(\mathcal{P}(\omega)) = \aleph_1$. $\square$

SILVER machines

Consider a structure $M = (\mathrm{Ord}, <, M)$, $M\colon \mathrm{Ord}^{<\omega} \rightharpoonup \mathrm{Ord}$. For $\alpha \in \mathrm{Ord}$ let

$$M^\alpha = (\alpha, <, M \cap \alpha^{<\omega});$$

for $X \subseteq \alpha$ let $M^\alpha[X]$ be the substructure of M^α generated by X.

M is a SILVER *machine* if it satisfies the following axioms:

– (*Condensation*) For $\alpha \in \mathrm{Ord}$ and $X \subseteq \alpha$ there is a unique β such that $M^\beta \cong M^\alpha[X]$;

– (*Finiteness property*) For $\alpha \in \mathrm{Ord}$ there is a finite set $z \subseteq \alpha$ such that for all $X \subseteq \alpha + 1$

$$M^{\alpha+1}[X] \subseteq M^\alpha[(X \cap \alpha) \cup z] \cup \{\alpha\};$$

– (*Collapsing property*) If the limit ordinal β is singular in L then there is $\alpha < \beta$ and a finite set $p \subseteq \mathrm{Ord}$ such that $M[\alpha \cup p] \cap \beta$ is cofinal in β.

Basic idea for turning an ordinal register machine into a Silver machine:

$$M(\ulcorner P \urcorner, l, \beta_0, ..., \beta_n) = \gamma \text{ iff } P(\vec{\beta}\,) = \vec{\gamma} \text{ and } \gamma \text{ is the } l\text{-th component of } \vec{\gamma}\,.$$

Define initial machines $M^{\vec{\alpha}}$ for ordinal sequences $\vec{\alpha} = (\alpha_0, \alpha_1, \ldots)$ by $M^{\vec{\alpha}}(\ulcorner P \urcorner, l, \beta_0, ..., \beta_n) = \gamma$ iff $M(\ulcorner P \urcorner, l, \beta_0, ..., \beta_n) = \gamma$ and during the computation by P the register content of the register `Rn` was always $< \alpha_n$.

For condensation, consider a transitivisation

$$\pi : (\beta, <) \cong (M^{\vec{\alpha}}[X], <).$$

We have to show that π respects computations, i.e.

$$P(\vec{\xi}\,) = \vec{\zeta} \text{ iff } P(\pi(\vec{\xi}\,)) = \pi(\vec{\zeta}\,).$$

Easy for the instructions `Rn:=0` and `Rn:=Rn+1`. Modify the programming language to have the only other instruction

```
while Rm < Rn {Instructions}.
```

We have to see inductively that this loop is respected by π if all the instructions in `{Instructions}` are respected.

This is clear for single traversals of the loop. If the loop is traversed λ times for λ a limit ordinal we have to see that the limit rule for register contents is respected by π. Using `while` instructions there will be enough witnesses to the limit behaviour in $M^{\vec{\alpha}}[X]$.

We can carry out SILVER's proof of $\Box$ with the resulting machine.

Can this proof be re-phrased (and better understood) in the language of ordinal register machines?

Let β be a singular limit ordinal.

Let α_0 be minimal, such that the singularity of β can be computed with all registers $< \alpha_0$.

Let $\alpha_1 \leqslant \alpha_0$ be minimal, such that the singularity of β can be computed with register $\mathtt{R0} < \alpha_0$ and all other registers $< \alpha_1$.

Let $\alpha_2 \leqslant \alpha_1$ be minimal, such that the singularity of β can be computed with registers $\mathtt{R0} < \alpha_0$, $\mathtt{R1} < \alpha_1$ and all other registers $< \alpha_2$.

...

???