Effectively Computable Ordinal Functions

Peter Koepke, University of Bonn

EMU 2009, CUNY, New York, August 18, 2009
Contents

- Primitive recursive set and ordinal functions
- Recursive set and ordinal functions
- Ordinal TURING machines
- Ordinal register machines
- The constructible model $L$
- Equivalences of the various computabilities
**Primitive recursive set (ordinal) functions (R. Jensen and C. Karp, R. Gandy)**

A function $F: V \rightarrow V$ ($F: \text{Ord} \rightarrow \text{Ord}$) is a *primitive recursive set (ordinal) function* iff it is generated by the following scheme:

- $P_{n,i}(\vec{x}) = x_i$, $1 \leq n \in \omega$, $\vec{x} = (x_1, \ldots, x_n)$, $1 \leq i \leq n$
- $F(x) = 0$
- $F(x, y) = x \cup \{y\}$ ($F(\alpha) = \alpha \cup \{\alpha\} = \alpha + 1$)
- $C(x, y, u, v) = x$ if $u \in v$, $= y$ otherwise
**Primitive Recursive Set (Ordinal) Functions**

- \( F(\vec{x}, \vec{y}) = G(\vec{x}, H(\vec{x}), \vec{y}) \)
- \( F(\vec{x}, \vec{y}) = G(H(\vec{x}), \vec{y}) \)
- Recursion:
  \[
  F(\vec{z}, \vec{x}) = G(\bigcup \{ F(u, \vec{x}) \mid u \in \vec{z} \}, \vec{z}, \vec{x})
  \]
**Set recursion**

\[ F(z, x) = G\left(\bigcup \{ F(u, x) | u \in z \}, z, x\right) \]

allows course-of-value recursion:

\[ F^* \upharpoonright \text{TC} (\{z\}) = \bigcup \{ F^* \upharpoonright \text{TC} (\{u\}) | u \in z \} \cup \bigcup \{ (z, G^* (\bigcup \{ F^* \upharpoonright \text{TC} (\{u\}) | u \in z \})) \} \]
ORDINAL RECURSION

\[ F(\alpha, \vec{x}) = G(\bigcup \{ F(\beta, \vec{x}) \mid \beta \in \alpha \}, \alpha, \vec{x}) \]
\[ = G(\lim_{\beta < \alpha} F(\beta, \vec{x}), \alpha, \vec{x}) \]

appears weaker: how can courses-of-values be coded into single ordinals?
R. Jensen and M. Schröder:

**Theorem.** Let \( F : \text{Ord} \to \text{Ord} \). Then \( F \) is primitive ordinal recursive iff \( F \) is primitive set recursive.

The **Proof** uses the constructible hierarchy.
The constructible hierarchy (Gödel)

- $L_0 = \emptyset$

- $L_{\alpha+1} = \text{Def}(L_\alpha) =$ the set of all subsets of $L_\alpha$ which are first-order definable in the structure $(L_\alpha, \in)$ from parameters

- $L_\lambda = \bigcup_{\alpha < \lambda} L_\alpha$, if $\lambda$ is a limit ordinal

- $L = \bigcup_{\alpha \in \text{Ord}} L_\alpha$ is the constructible universe
The constructible hierarchy

$L = \bigcup L_\alpha$
**The Constructible Hierarchy**

Every element of $L$ is of the form

$$x_0 = \{u_0 \in L_{\alpha_0} \mid L_{\alpha_0} \models \varphi_0(u_0, x_1, \ldots)\}$$

$$= \{u_0 \in L_{\alpha_0} \mid L_{\alpha_0} \models \varphi_0(u_0, \{u_1 \in L_{\alpha_1} \mid L_{\alpha_1} \models \varphi_1(u_1, x_2, \ldots)\}, \ldots)\}$$

$$= \{u_0 \in L_{\alpha_0} \mid L_{\alpha_0} \models \varphi_0(u_0, \{u_1 \in L_{\alpha_1} \mid L_{\alpha_1} \models \varphi_1(u_1, \{u_2 \in L_{\alpha_2} \mid L_{\alpha_2} \models \varphi_2(u_2, x_3, \ldots)\}, \ldots)\}, \ldots)\}$$

$$= \ldots$$

and can be “named” by a finite sequence of ordinals like

$$\alpha_0, \varphi_0, \alpha_1, \varphi_1, \alpha_2, \varphi_2, \ldots$$
THE CONSTRUCTIBLE HIERARCHY

Finite sequences of ordinals can be coded by single ordinals due to Gödel pairing functions: there are primitive recursive ordinal functions $G, G_1, G_2$ such that

- $G : \text{Ord} \times \text{Ord} \leftrightarrow \text{Ord}$

- $\forall \alpha \ G(G_1(\alpha), G_2(\alpha)) = \alpha$

The basic operations for the (coded) constructible universe are primitive recursive ordinal functions (Takeuti; Jensen, Schröder)
Recursive ordinal functions (Jensen, Karp)

A function $F: V \rightarrow V$ ($F: \text{Ord} \rightarrow \text{Ord}$) is a set (ordinal) recursive function iff it is generated by the above scheme together with the minimisation rule

$$F(x) = \min \{ \xi | G(\xi, x) = 0 \}, \text{ provided that } \forall x \exists \xi G(\xi, x) = 0$$
Recursive ordinal functions

Theorem. For $f: \text{Ord} \rightarrow \text{Ord}$ the following are equivalent:

- $f$ is ordinal recursive
- $f$ is set recursive
- $f$ is $\Delta_1(L)$
TURING machines

Turing Program

Time \( \omega \)

Turing head

Tape of length \( \omega \)

0 1 1 0 1 0 1 1 1 ...
Ordinal Turing machines (OTMs)
Ordinal Turing machines (OTMs)

- successor steps of computations are determined by standard commands:
  
  \[ m: \text{if read}=0 \ (\text{or } 1) \ \text{then write } 0 \ (\text{or } 1), \ \text{go right (or left), and jump to instruction } n \]

- limit steps \( \lambda \) are determined by liminf’s:
  
  \[ \begin{align*}
  \text{command}(\lambda) &= \liminf_{\alpha < \lambda} \text{command}(\alpha) \\
  \text{head}(\lambda) &= \liminf_{\alpha < \lambda} \text{head}(\alpha) \\
  \text{cell}_\gamma(\lambda) &= \liminf_{\alpha < \lambda} \text{cell}_\gamma(\alpha)
  \end{align*} \]
OTM Computability $\iff$ constructibility

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

**Proof.** ($\rightarrow$) Any OTM computation can be carried out inside the model $L$, hence $X \in L$. 
The following OTM algorithm computes all constructible sets: assume that a structure $(X, R)$ is written on the tape which is (pre-)isomorphic to $(L_\alpha, \in)$. Extend $(X, R)$ to a structure $(X', R')$ (pre-)isomorphic to $(L_{\alpha+1}, \in)$: for each $\in$-formula $\varphi(v_0, v_1, \ldots, v_m)$ and $x_1, \ldots, x_n \in X$ pick a new point $z \in X' \setminus X$ and for $x_0 \in X$ let

$$x_0 \overset{R'}{\to} z \text{ iff } (X, R) \models \varphi[x_0, x_1, \ldots, x_m]$$

Every constructible set of ordinals occurs in the construction and is hence OTM computable.
Total functions $\text{Ord} \rightarrow \text{Ord}$

**Theorem** (K, B. Seyfferth) $f: \text{Ord} \rightarrow \text{Ord}$ is OTM computable iff $f$ is $\Delta_1(L)$.

**Proof.** ($\rightarrow$) Let $f$ be computable by the program $P$.

$$f(\alpha) = \beta \iff \exists \text{computation } C \text{ according to } P \text{ with input } \alpha \text{ and output } \beta$$

$$\iff \exists \text{computation } C \in L \text{ according to } P \text{ with input } \alpha \text{ and output } \beta$$

$$\iff \forall \text{computation } C \in L (\text{if } C \text{ is according to } P \text{ with input } \alpha \text{ then } C \text{ outputs } \beta$$
Let \( f : \text{Ord} \rightarrow \text{Ord} \) be defined in \((L, \in)\) by the \(\Sigma_1\)-formula \(\varphi(x, y)\). Then compute \(f(\alpha)\) as follows: enumerate \(L\) as described above. In the enumeration search for some structure \((X, R)\) and \(x, y \in X\) such that \((X, R) \models \varphi(x, y)\) and \(\text{otp}_R(x) = \alpha, \text{otp}_R(y) = \beta\).
Register machines

\[ R_0 \in \omega \]
\[ R_1 \in \omega \]
\[ \vdots \]
\[ R_n \in \omega \]

Control
Program
States

Time
0 1 ... \( \omega \)
Ordinal register machines, (ORMs)

- $R_0 \in \text{Ord}$
- $R_1 \in \text{Ord}$
- $\vdots$
- $R_n \in \text{Ord}$

Time:

\[ 0, 1, \omega, \omega + 1, \aleph_1 \]
ORM Computability $\iff$ constructibility

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

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

($\leftarrow$) Since $L$ has a canonical wellordering every point $x \in L$ can be “named” by a single ordinal $\alpha$; $x$ is the “interpretation” $I(\alpha)$ of the name $\alpha$. To compute $\Sigma_0$-properties of $I(\alpha)$ one suffices to compute $\Sigma_0$-properties of sets $I(\alpha')$ with $\alpha' < \alpha$. This amounts to a recursion which can be organised by a stack. Such stacks can be emulated by ORMs.
A recursion theorem

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

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

Then $F : \text{Ord} \rightarrow \text{Ord}$ is ORM computable.
A recursion theorem

\[ F(\alpha) = 1 \text{ iff } \exists \beta < \alpha \ H(\alpha, \beta, F(\beta)) = 1 \]
A recursion theorem

\[ F(\alpha) \]

\[ F(0) \quad F(1) \quad \ldots \quad F(\beta) \]

\[ F(0) \quad F(0) \quad F(1) \quad \ldots \quad F(\gamma) \]

\[ F(0) \quad F(\delta) \]

\[ \vdots \quad \vdots \]
A recursion theorem

Search for a good path using a stack $F(\alpha)$, $F(\beta)$, $F(\gamma)$, ...
A recursion theorem

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

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

$L = \bigcup L_\alpha$

$L_{\alpha+1}$

$L_\alpha$

$F(\alpha) = 1$

iff $\exists \beta < \alpha \ H(\alpha, \beta, F(\beta)) = 1$
Total functions $\text{Ord} \rightarrow \text{Ord}$

**Theorem (K)** $f: \text{Ord} \rightarrow \text{Ord}$ is ORM computable iff $f$ is $\Delta_1(L)$.

**Proof.** $(\rightarrow)$ Let $f$ be computable by the program $P$.

$$f(\alpha) = \beta \text{ iff } \exists \text{computation } C \text{ according to } P \text{ with input } \alpha \text{ and output } \beta$$

$$\text{iff } \exists \text{computation } C \in L \text{ according to } P \text{ with input } \alpha \text{ and output } \beta$$

$$\text{iff } \forall \text{computation } C \in L(\text{if } C \text{ is according to } P \text{ with input } \alpha \text{ then } C \text{ outputs } \beta$$
Let \( f : \text{Ord} \rightarrow \text{Ord} \) be defined in \((L, \in)\) by the formula \( \exists z \psi(x, y, z) \) where \( \psi \) is \( \Sigma_0 \). Then compute \( f(\alpha) \) as follows: compute a “name” \( \dot{\alpha} \) for \( \alpha \); search for ordinals \( \dot{\beta} \) and \( \dot{\gamma} \) such that \( \psi(\alpha, I(\dot{\beta}), I(\dot{\gamma})) \); if such \( \dot{\beta}, \dot{\gamma} \) are found, compute and output \( \beta = I(\dot{\beta}) \).
Theorem. For $f: \text{Ord} \rightarrow \text{Ord}$ the following are equivalent:

- $f$ is recursive à la Jensen and Karp
- $f$ is $\Delta_1(L)$
- $f$ is OTM computable
- $f$ is ORM computable
The Church-Turing thesis according to Odifreddi

For $f: \omega \to \omega$ the following are equivalent:

- $f$ is recursive
- $f$ is finitely definable
- $f$ is Herbrand-Gödel computable
- $f$ is representable in a consistent formal system extending $\mathcal{R}$
The Church-Turing thesis according to Odifreddi

For $f: \omega \to \omega$ the following are equivalent:

- $f$ is recursive
- $f$ is flowchart (or “while”) computable
- $f$ is $\lambda$-computable
**Theorem.** For \( f : \text{Ord} \to \text{Ord} \) the following are equivalent:

- \( f \) is recursive à la Jensen and Karp
- \( f \) is \( \Delta_1(L) \)
- \( f \) is OTM computable
- \( f \) is ORM computable
- \( f \) is “while” computable on the ordinals
- \( f \) is computable by the methods of Kripke, Platek, Machover, Takeuti
Conclusion

There is a stable and well-characterised notion of effectively computable ordinal function.